

Modelagem de Informações Baseadas
em Objetos Aplicados às Fases do
Processo de Projeto de Produto

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Programa de Pós-Graduação em Engenharia Mecânica

Modelagem de Informações Baseadas em Objetos
Aplicados às Fases do Processo de Projeto de Produto

Tese submetida à Universidade Federal de Santa
Catarina para a obtenção do grau de doutor em
Engenharia Mecânica

Raimundo Ricardo Matos da Cunha

Florianópolis, Abril de 2004.

Biografia

Raimundo Ricardo Matos da Cunha é natural de Fortaleza-CE. Engenheiro Mecânico pela Universidade Federal do Ceará – UFC. Mestre em Engenharia Mecânica pela Universidade Federal de Santa Catarina – UFSC. Atualmente é professor universitário e leciona disciplinas de Desenho Técnico, Desenho Auxiliado por Computador, Projeto e Manufatura Auxiliada por Computador, além de outras da área de projeto mecânico do currículo de Engenharia.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Programa de Pós-Graduação em Engenharia Mecânica

Modelagem de Informações Baseadas em Objetos
Aplicados às Fases do Processo de Projeto de Produto

Raimundo Ricardo Matos da Cunha

Esta tese foi julgada adequada para a obtenção do título de
Doutor em Engenharia
Especialidade Engenharia Mecânica, sendo aprovada em sua
forma final.

Altamir Dias, D.Sc.
Orientador

José Antônio Bellini da Cunha Neto, Dr.
Coordenador do Programa de Pós-graduação em Engenharia Mecânica

BANCA EXAMINADORA:

Altamir Dias, D.Sc. – Presidente

Carlos Alberto Costa, Ph.D. – Membro-Relator

Andy Solly Segenreich, Ph.D. – Membro

André Ogliari, D. Eng., D. Eng. – Membro

Ilson Wilmar Rodrigues Filho, D. Eng. – Membro

Epígrafe

“Antes de ensinar, o homem está aqui para aprender junto.”

*“A vida nos mostra as deficiências, fraquezas e limitações da raça humana.
Saber identificá-las, experimentá-las e apreendê-las é uma virtude para si.
Mas a maior virtude é saber disso e convertê-las para o bem.”*

Autores desconhecidos

Dedicatória

À minha família,
Ao início de vida das minhas filhas, Catharina e Sophia,
À minha avó Maria de Lourdes de Sousa Matos (*In memoriam*).

Agradecimentos

A Deus, pela saúde e força interior que me concedeu.

Ao CNPq, pela assistência financeira.

À Universidade Federal de Santa Catarina (UFSC) e ao Programa de Pós-graduação em Engenharia Mecânica (POSMEC), pelo apoio logístico.

Aos membros da banca examinadora, pelas muitas contribuições na forma de objeções e sugestões para esse trabalho chegar à forma final.

A todos os colegas de laboratório e estudo que compartilharam comigo esses dias, na forma de alegrias, idéias, discussões, comentários, brincadeiras, ensinamentos, aprendizados, críticas, e fraternidade.

Sumário

LISTA DE FIGURAS	X
LISTA DE TABELAS	XII
LISTA DE SIGLAS	XIII
RESUMO	XIV
ABSTRACT	XVI
CAPÍTULO 1 – APRESENTAÇÃO DA TESE	1
1.1. INTRODUÇÃO	1
1.2. CONTEXTO E MOTIVAÇÕES DA TESE	1
1.2.1. Questões Associadas ao Contexto da Tese	5
1.2.1.1. CAD com Modelagem de Features e Orientação a Objetos e a Validação de Features	6
1.2.1.2. Normalização das Terminologias e Conceitos Associados.....	7
1.3. ESCOPO DA TESE	8
1.4. OBJETIVOS.....	10
CAPÍTULO 2 – PROCESSO DE PROJETO DE PRODUTO E SISTEMAS CAD	11
2.1. INTRODUÇÃO	11
2.2. PROCESSO DE PROJETO	13
2.3. A INFORMAÇÃO NO PROJETO	16
2.4. SISTEMAS CAD E O PROCESSO DE PROJETO.....	23
2.4.1. Mudanças na Aplicação do Sistema CAD.....	24
2.4.2. Recursos Computacionais dos Sistemas CAD.....	28
CAPÍTULO 3 – OBJETOS E FEATURES DE PROJETO.....	32
3.1. INTRODUÇÃO	32
3.2. CONCEITOS E DEFINIÇÕES	33
3.2.1. Informação e Dado	33
3.2.2. Conceito de Objeto e Feature.....	36
3.3. CONSIDERAÇÕES SOBRE AS PROPOSIÇÕES DA TESE	44
CAPÍTULO 4 – SISTEMÁTICA DE FEATURIZAÇÃO NO PROCESSO DE PROJETO DE PRODUTO	46
4.1. INTRODUÇÃO	46
4.2. CONCEITOS E DEFINIÇÕES	47
4.3. SISTEMÁTICA DE FEATURIZAÇÃO.....	49
4.4. ESTUDO DE CASO.....	53
4.5. CONSIDERAÇÕES PARA A TESE.....	58

CAPÍTULO 5 – FEATURES NAS FASES DO PROCESSO DE PROJETO DE PRODUTO	59
5.1. INTRODUÇÃO	59
5.2. TRANSFORMAÇÃO DOS DADOS DE PROJETO	60
5.3. FLUXO DE INFORMAÇÃO NO PROJETO	62
5.4. IDENTIFICAÇÃO E CLASSIFICAÇÃO DAS FEATURES DE PROJETO	66
5.4.1. Caracterização das Features de Projeto	66
5.4.2. Critérios de Definição dos Atributos das Features	67
5.4.3. A Hierarquia de Features	68
5.5. CONSIDERAÇÕES.....	70
CAPÍTULO 6 – RESULTADOS: IMPLEMENTAÇÃO DOS OBJETOS E FEATURES DE FASE DE PROJETO	72
6.1. INTRODUÇÃO	72
6.2. FEATURES COMO UNIDADE DE INFORMAÇÃO PARA OBJETOS DE PROJETO	74
6.2.1. Classes da Fase de Projeto Informacional	75
6.2.2. Classes da Fase de Projeto Conceitual	81
6.2.3. Classes da Fase de Projeto Preliminar e Detalhado	84
6.3. ESTUDO DE CASO.....	86
6.3.1. Exemplificando no Projeto Informacional.....	87
6.3.2. Exemplificando no Projeto Conceitual	89
6.3.3. Exemplificando no Projeto Preliminar e Detalhado.....	91
CAPÍTULO 7 – CONCLUSÕES E CONSIDERAÇÕES FINAIS.....	95
7.1. CONCLUSÕES E CONSIDERAÇÕES DA TESE	95
7.2. DIFICULDADES DO PROCESSO DE MODELAGEM	97
7.3. SUGESTÕES PARA TRABALHOS FUTUROS	98
REFERÊNCIAS BIBLIOGRÁFICAS	100
BIBLIOGRAFIAS	104
APÊNDICE 1 – NOTAÇÃO E SIMBOLOGIA UML	109
APÊNDICE 2 – PESQUISA SOBRE O CONCEITO DE FEATURE.....	111

Lista de Figuras

Figura 1. Sistema de fabricação e atividades externas de produção.	12
Figura 2. Sistemática do processo de projeto segundo Pahl e Beitz.	14
Figura 3. Modelo do processo de projeto segundo Oshuga.	15
Figura 4. Visão do desenvolvimento do produto durante o projeto informacional.	18
Figura 5. Seqüência de processamento das informações de projeto e transformação dos dados no projeto informacional.	18
Figura 6. Matriz de levantamento de requisitos de usuário.	20
Figura 7. Matriz para conversão dos requisitos de usuário em requisitos de projeto.	20
Figura 8. Exemplo de questionário estruturado para levantamento de necessidades.	21
Figura 9. Estrutura de função do produto numa forma hierárquica e representada por ícones.	22
Figura 10. Exemplo de matriz morfológica para representação das alternativas de solução do problema.	22
Figura 11. Evolução da modelagem geométrica dos modelos em sistemas CAD.	26
Figura 12. Exemplo de perfis bidimensionais gerando formas curvas e sinuosas.	29
Figura 13. Biblioteca de componentes padronizados do MicroStation/J Modeler v7.1.	30
Figura 14. Biblioteca de componentes padronizados da MechSoft, Inc.	30
Figura 15. O processo de transformação de dados em informação.	34
Figura 16. O processo de transformação de dados em informação no projeto.	36
Figura 17. Convergência entre os contextos computacionais de modelagem orientada a objetos e o de projeto baseado em <i>features</i> .	39
Figura 18. Enquadramento da visão de <i>feature</i> de projeto na tese.	45
Figura 19. Estrutura hierárquica da informação do produto no sistema CAD.	49
Figura 20. Os passos da <i>featurização</i> do produto.	50
Figura 21. O produto e sua especificação técnica e o conjunto estudado.	54
Figura 22. Peça e perfis parametrizados na geração de formas usando <i>features</i> .	55
Figura 23. Biblioteca de perfis do conjunto coluna maior da betoneira 120L.	55
Figura 24. Fases de projeto com as respectivas entradas e saídas.	61
Figura 25. Fases de projeto e documentação do produto.	61
Figura 26. Processos x Fases de Projeto – Produto x Dados e Informações.	62
Figura 27. Fluxo de informações e a representação usada nas diferentes fases do PPP	65
Figura 28. Primeira proposta de modelagem das <i>features</i> de fase do PPP.	68
Figura 29. Hierarquia de <i>features</i> no processo de projeto.	69
Figura 30. Hierarquia de <i>features</i> .	71
Figura 31. Classes do contexto do PPP.	73

Figura 32. Diagrama UML de classes do PPP.	74
Figura 33. Diagrama UML de classes do projeto informacional.	80
Figura 34. Diagrama UML de classes do projeto conceitual.	83
Figura 35. Especificação da geometria de um produto.	84
Figura 36. Diagrama UML de classes do projeto preliminar e detalhado.	86
Figura 37. Casa da qualidade do aparelho portátil de hidromassagem.	89
Figura 38. Estrutura funcional do aparelho portátil de hidromassagem.	90
Figura 39. Estrutura funcional do aparelho portátil de hidromassagem em UML.	91
Figura 40. Croqui do leiaute do sistema de aquecimento da água do aparelho portátil de hidromassagem.	92
Figura 41. Vista explodida do sistema de aquecimento da água.	92
Figura 42. Resistor	93
Figura 43. Contactador	93
Figura 44. Isolante	93
Figura 45. Mola	93
Figura 46. Encosto anterior	93
Figura 47. Encosto posterior	93
Figura 48. A montagem do sistema de aquecimento.	94

Lista de Tabelas

Tabela 1. Categorias de informação na fase de projeto informacional.	19
Tabela 2. Comparações entre dado de projeto e informação de projeto.	36
Tabela 3. Exemplo da geração de formas das peças com restrições e significados semânticos associados e encapsulados pela sistemática de <i>featurização</i> .	56

Lista de Siglas

API – Application Programming Interface.....	9
CAD – Computer-Aided Design	3
CASE – Computer-Aided Software Engineering	98
CAX – Computer-Aided "X"	3
DBMS – Database Management System	23
ER – Entidade / Relacionamento	98
JDBC – Java Database Connectivity	30
JMDL – Java MicroStation Developer Language	30
MDL – MicroStation Developer Language	30
ODBC – Open Database Connectivity.....	30
PPP – Processo de Projeto de Produto.....	2
QFD – Quality Function Deployment.....	88

Resumo

As informações de projeto possuem características de natureza geométricas e não-geométricas. Estes dois tipos de informação são compostos por dados, os quais são agrupados numa estrutura de dados única – as *features* de projeto – para terem um significado de Engenharia. Esta tese utiliza o conceito de *features* de projeto e o paradigma de orientação a objetos para fazer a identificação de informações de cada uma das fases do projeto mecânico, a saber: Projeto Informacional (PI), Projeto Conceitual (PC), Projeto Preliminar (PP), e Projeto Detalhado (PD); e assim definir uma assinatura de dados, relacionamentos e operações de cada fase do projeto. Além disso, o conjunto de *features* de projeto de uma fase posterior deve ser uma evolução da informação geométrica e não-geométrica de uma fase anterior.

Os objetivos principais da tese são: definir e modelar classes de *features* que encapsulem os atributos relevantes para capturar e caracterizar a evolução e transformação das informações e dados das fases do processo de projeto; contextualizar os conceitos de *features* de projeto e objetos, ressaltando similaridades e diferenças; e proporcionar o reuso, a troca e o compartilhamento das informações de projeto através de uma base genérica de classes de *features* de projeto. As *features* identificadas e definidas servem de informação para o projetista e documentam o processo de projeto de produto.

A abordagem inicial consiste na identificação e classificação de *features* que evoluam nas diferentes fases do projeto mecânico de produtos. Para isso foram implementadas as classes de *features* próprias de cada fase do projeto enumerando a informação não-geométrica necessária para dar um significado de aplicação. Definidas as classes de *features das fases de projeto* fica possível através de biblioteca de funções de um sistema CAD a inserção das *features* para dentro do ambiente de modelagem de um sistema CAD convencional, a fim de criar o projeto detalhado do produto também com informações de outras fases do processo de projeto de produto.

A pesquisa contribui para enfatizar a necessidade de mudança na forma de pensar a estrutura de dados em que um sistema computacional deve ser baseado e implementado para servir de suporte no processo de projeto produto. Ela também contribui para uma integração maior de sistemas computacionais às fases informacional, conceitual, preliminar e detalhada da sistemática de projeto. Além de comprovar que o paradigma de orientação a objetos e o conceito de *feature* de projeto são os modelos mais adequados e modernos para essa modelagem de informações de Engenharia.

Abstract

The product development process includes geometric and non-geometric information as well as design information. These properties are generally called data, but here, it is thought more abstractly as design features. This research proposes the concept of design features and it uses the object-oriented paradigm to model and define such features. Those features are organized based on to design process phases commonly used in mechanical design, namely: Informational Design, Conceptual Design, Preliminary Design, and Detailed Design. The goal is to identify a phase signature for each design phase, its relationships with others phases and the attributes and methods necessary to get the data and information in the whole design process. Besides, the set of features of the current phase should be an evolution of the geometric and non-geometric information of the previous phase.

The main objective is to model design features that encapsulate the evolution of the data and to capture Engineering information in the design phases. Also is necessary to capture the intention of the designer when using the defined features and implemented in the CAD system. Finally to propose a library of features that able the reusing, editing and sharing design information in a computational environment.

The research concentrates on the study of the features technology in a design point of view focused to design and to object-oriented modeling. The initial approach consists of the features identification and classification thought in the different phases of the product mechanical design. For each phase, a set of features were identified and implemented. The detailed design features are conceived related to functions library used by CAD systems. It goes in the direction to converge to CAD system features present in modeling environment, in order to create the detailed design with much more complete product information together other information phases of the design process.

This research contributes to emphasize the change necessary in the

way of thinking the data structure that design process should be developed. Besides, it shows that the object-oriented paradigm and the concept of design feature is the most appropriate and modern models for information modeling of design process. The identified and defined features in that abstraction domain serve as information for the designer and they document the design process. Besides, it contributes to a better integration among phases of the design process methodology and a computational system.

Capítulo 1

Apresentação da Tese

1.1. INTRODUÇÃO

Este capítulo faz uma apresentação geral da tese, para tornar mais claro os relacionamentos entre os assuntos abordados.

O objetivo principal é evidenciar o escopo de atuação – quais são as principais áreas de conhecimento da tese – concentrando o estado da arte e as contribuições desse trabalho.

Na seqüência, são descritas as motivações que fundamentaram à escolha do tema de pesquisa, citando quais são os objetivos e as contribuições possíveis da implementação desta tese.

Ao longo do texto também são citados os trabalhos de referência que se alinham e que deram suporte às idéias descritas na tese.

1.2. CONTEXTO E MOTIVAÇÕES DA TESE

Algumas questões, baseadas em discussões e leitura de referências bibliográficas sobre o tema, motivaram a pesquisa. Numa maneira de iniciar e incitar a discussão, as principais questões são listadas a seguir:

1. *Como os dados e informações de projeto evoluem e se transformam?*
2. *Quais são os dados e informações de projeto que podem ser identificados e modelados?*
3. *De quais formas os dados e informações de projeto podem ser integrados e capturados num sistema computacional durante o processo de projeto?*
4. *Como e o que é preciso, em termos computacionais, para propiciar e estender funcionalidades de suporte à atividade de projetar?*
5. *Quais modelos de informação são mais adequados e atuais nessa área de modelagem de informações de suporte ao processo de projeto?*

As questões acima serão respondidas e comentadas ao longo do texto dessa tese, sendo oportunamente referenciadas. A partir dessas questões levantadas, é possível concluir que as necessidades atuais de mercado e o progresso das pesquisas na área de projeto mecânico resumem-se a três importantes níveis de pesquisa, os quais foram considerados nesta tese:

- Um nível básico, relacionado à questão da **modelagem computacional** das informações de projeto. Numa linguagem computacional de modelagem do problema, esse nível corresponde à Fase de Análise.
- Um nível intermediário, relacionado à **representação computacional** das informações de projeto. Esse nível corresponde à Fase de Projeto.
- E um nível final, relacionado à questão da **integração de ferramentas computacionais** de suporte ao PPP – Processo de Projeto de Produto. Esse nível corresponde à Fase de Uso / Utilização do sistema computacional.

Nessa tese, os níveis básico e intermediário consistem na modelagem e representação de dados de projeto baseados em *features* de projeto e implementadas conforme os princípios de orientação a objetos. Essa abstração de níveis é requisito básico para consolidação de uma forma renovada de como desenvolver e estruturar ferramentas computacionais de apoio ao projeto, contribuindo para o estado da arte em sistemas

computacionais de suporte ao processo de projeto e desenvolvimento de produtos como um todo, entre os quais citamos os Sistemas CAD – *Computer-Aided Design*.

Já o nível final é entendido como a maneira de implementação da aplicação e uso das decisões tomadas nos dois primeiros níveis – de modelagem e de representação.

Na tese, a modelagem está focalizada em proporcionar as bases para a captura, armazenamento e a evolução das informações nas fases de projeto, contribuindo para o estado da arte da modelagem computacional de informações do PPP. E o resultado é a identificação de uma base genérica de informações na forma de classes de *features* orientadas a objeto.

Essas áreas de conhecimento e as linhas de pesquisa, por sua vez, buscam suporte numa outra diversidade de áreas de pesquisas relacionadas ao projeto mecânico, tais como: metodologias de projeto, desenvolvimento de produto, e implementação de ferramentas computacionais de suporte ao PPP. Atualmente, existem temas em pesquisas para aplicação em Sistemas CAD e CAX's¹ (*Computer-Aided "X"*), nas sub-áreas de modelagem de informações de projeto, tecnologia de *features*, e padrões para troca de dados de Projeto (OSHUGA, 1989; SHAH e MATHEW, 1991, MÄNNISTO *et al.*, 1998; NYQVIST e NIELSEN, 2001). As duas primeiras sub-áreas citadas acima estão dentro do escopo de pesquisa dessa tese.

Em razão disso, alguns fatores diretos e indiretamente relacionados motivam desenvolver pesquisas dentro desse contexto, e convencem da possibilidade de contribuir para o estado da arte aplicado ao suporte computacional do PPP:

- Com base na bibliografia de *features* e orientação a objeto referenciada no texto desta tese, existem poucas discussões e/ou argumentos levando em conta o significado e/ou a interpretação das *features* no contexto de projeto. Além do que a verificação de uma relação de equivalência entre os conceitos de objeto e *features* no contexto de representação computacional da informação. Apesar dos

¹ A sigla CAX contempla as várias siglas que referenciam as tecnologias computacionais que auxiliam atividades de Engenharia e que são baseadas no computador, a exemplo de CAD, CAE, CAM, CAPP, CAQ, CAFD, etc..

anos de pesquisa em tecnologia de *features*, desde quando surgiu o conceito em (CUNNINGHAM e DIXON, 1988), e a aplicação da modelagem orientada a objetos, ainda hoje esses paradigmas não foram inteiramente integrados numa estrutura de dados única. O principal motivo advém do amadurecimento ainda incipiente em tratar e discutir ambos os conceitos num mesmo domínio de abstração da modelagem, seja ou no de orientação a objeto, ou no de tecnologia de *features*. O Capítulo 3 inicia uma discussão sobre a relação de equivalência entre os conceitos de objetos e *features*, com o objetivo de contribuir para o esclarecimento dos conceitos básicos dessa proposta e incitar outras opiniões e pontos de vista;

- Principalmente nas fases iniciais do processo de projeto verifica-se a existência de estruturas de dados e interfaces computacionais inadequadas para lidar com a informação e os dados de projeto mais subjetivos e/ou não-geométricos, os quais não possuem uma conotação de valor ou quantidade, mas apenas de teor qualitativo e abstrato. Vários autores tiveram essa conclusão: (ULLMAN *et al.*, 1990), (HSU e WOON, 1998), e (OGLIARI, 1999). Em suma, está se falando das informações que geralmente são processadas na mente do projetista (*denominadas como “intenções de projeto”*), mas que não são nem estão refletidas e capturadas no modelo de informação do projeto ou produto, e que por isso não são compartilhadas ou reusadas; e muito menos implementadas nas aplicações e ferramentas computacionais. No Capítulo 2 descreve-se algumas formas de representação computacional das informações de projeto, e no Capítulo 6 tem-se a proposta de classes de *features* das fases de projeto que capturam as informações nas fases;
- Necessidade de integração entre ferramentas computacionais, que favoreçam a reusabilidade e interoperabilidade de informações de projeto, e por isso a importância de buscar um modelo que favoreça e possibilite a implementação dessa necessidade.

Esta tese direciona-se para um melhoramento da representação computacional das informações, focalizada no contexto do processo de projeto de produto, numa linha de pensamento já levantada por (WARMAN, 1990) – **A Modelagem de Objetos no Sistema CAD**. Dessa forma, estar-se-á preparando uma base de modelagem computacional sólida e consistente, com as vantagens que a orientação a objetos pode oferecer. As atividades subseqüentes do ciclo de vida do produto podem usar toda implementação realizada, e reusá-la noutro contexto de aplicação, seja de Planejamento do Processo (*projeto de sistemas de fixação de peças*), Fabricação (*geração do código comando numérico*), Análise de Tensão (*geração de malhas para elementos finitos*), Análise de Montabilidade, ou noutras atividades do ciclo de vida do produto.

Na literatura científica existem diversos trabalhos que discutem temas relacionados ao escopo dessa pesquisa. Todos acrescentam contribuições e conseguem avanços em pontos isolados ou específicos. Geralmente, a contribuição resulta num protótipo computacional que implementa as idéias defendidas pela pesquisa. Na revisão bibliográfica, esses trabalhos são discutidos com mais detalhes.

1.2.1. QUESTÕES ASSOCIADAS AO CONTEXTO DA TESE

Para a implementação da proposta de evolução dos dados de projeto, considerações relacionadas à modelagem de *features* e orientação a objetos nos sistemas CAD são relevantes nesta pesquisa. A partir desse estudo, deve-se ponderar sobre duas questões: a validação do modelo de *features* no CAD, e a padronização dos termos e definições para as *features* implementadas. Essas questões ainda não estão implementadas nos atuais sistemas CAD comerciais do mercado.

Nos sub-itens dessa secção, são feitos alguns comentários sobre essas questões associadas ao escopo dessa pesquisa de tese, e que são consideradas de suma relevância para a melhoria e aprimoramento da modelagem geométrica e também não-geométrica em Sistemas CAD comerciais futuros.

1.2.1.1. CAD COM MODELAGEM DE FEATURES E ORIENTAÇÃO A OBJETOS E A VALIDAÇÃO DE FEATURES

Em termos de desenvolvimento de sistemas computacionais e modelagem de problemas, segundo (WARMAN, 1990) a orientação a objetos é a tendência atual em sistemas CAD. E esta tendência vem acompanhada da modelagem de *features*, a qual se traduz no correspondente paradigma para fins de modelagem computacional das entidades geométricas e dos conceitos discutidos nessa proposta.

Sabe-se que, no nível mais básico, a representação das intenções de projeto e a implementação dos recursos computacionais de sistemas CAD recaem em operações sobre primitivas geométricas de baixo nível, tais como manipulação de pontos, linhas e curvas. Através dos recursos de personalização e programação busca-se aumentar o nível semântico das entidades geométricas pela introdução de algoritmos que capturem o raciocínio do projetista na modelagem de definição das entidades geométricas do sistema CAD. Tenta-se dessa forma, facilitar a interface de trabalho do usuário em manipular as entidades geométricas do CAD, trazendo-a para um nível de aplicação mais amigável e próximo do contexto.

Atualmente, o conceito mais evoluído, em termos de abstração de entidade para um sistema CAD, é o de *feature*. O conceito de *feature*, sob o ponto de vista tecnológico de Engenharia, tem um significado e uma relação direta com o encapsulamento de dados geométricos, topológicos e funcionais. O paradigma de orientação a objetos acrescenta ao conceito de *feature*, a semântica (*significado*) da intenção do projetista, representada pela identificação da função global do produto ou da forma (*geometria*, *topologia*), pelo nível de abstração, etc..

Através da parametrização e pela definição de restrições geométricas e relações dimensionais sobre os elementos gráficos que compõem as formas geométricas, tem-se a possibilidade de especificar uma *feature* ou objeto do domínio de Engenharia.

Todavia, quanto à definição de *features*, o que se tem nos atuais sistemas CAD são implementações estáticas, que não agregam mecanismos ou algoritmos de validação à definição da *feature*. Os algoritmos de validação garantem a integridade semântica das *features* durante as operações de

modelagem de interações com outras *features* já existentes no modelo. A importância da validação semântica para a modelagem de *features* pode ser constatada nos exemplos apresentados nos trabalhos de (SHAH e MÄNTYLÄ, 1995) e (BIDARRA, 1999).

É neste ponto, de validação das inter-relações entre *features* e manutenção do significado, que os sistemas CAD comerciais falham. A causa se deve à falta de algoritmos de validação para avaliar a integridade da semântica das *features*. Essa semântica das *features* depende da intenção do projetista em usar certa entidade geométrica dentro de seu contexto específico de aplicação. (BIDARRA e BRONSVOORT, 2000) advertem para o fato de que os modelos de *features* atualmente em uso nos sistemas CAD comerciais são mal definidos, pois ainda não adicionaram a validação semântica e a manutenção do significado dentro da definição da *feature*. Neste caso, o usuário fica encarregado de realizar a verificação e validação do modelo de *features* de acordo com a sua interpretação, ou de implementar a validação dentro da definição da *feature*.

1.2.1.2. PADRONIZAÇÃO DE TERMINOLOGIAS E CONCEITOS ASSOCIADOS

No caso da padronização de termos e conceitos utilizados durante o processo de projeto, a linha de pesquisa dessa proposta inicia um estudo que deve modelar os principais conceitos e entidades de projeto. É um importante benefício resultante da aplicação de uma base comum de *features*, conceitos e funções de projeto. Essa base comum minimiza e unifica o escopo de interpretações por parte dos integrantes de uma equipe de projeto.

Conclui-se também que a maior dificuldade em qualquer classificação de *features*, passa pela questão da falta de uma padronização das representações computacionais utilizadas. Ocorre que, normalmente, embora as *features* sejam definidas numa mesma semântica, elas podem apresentar diferentes representações computacionais. A padronização não é uma tarefa fácil em qualquer área do conhecimento científico. As variáveis que influenciam a aceitação ou não de uma padronização são muitas.

1.3. ESCOPO DA TESE

Pelo que foi introduzido na secção 1.2 – Contexto e Motivações da Tese – o escopo desse trabalho abrange dois itens principais, os quais se constituem na maior parte da discussão colocada no texto, a saber:

- A modelagem de informações baseadas numa abordagem orientada a objetos e modeladas como *features* para as fases do projeto mecânico de produtos;
- O suporte computacional na evolução das informações e transformação dos dados de projeto mecânico de produtos durante as fases de projeto.

Em outras palavras, essa tese é focalizada na modelagem orientada a objetos das informações das fases de projeto, colocando-as na forma de uma representação de classes de objetos que modelam as *features* de projeto.

As informações no decurso do projeto podem ser caracterizadas pelo sua natureza geométrica e não-geométrica. Estes dois tipos de informação precisam ser encapsulados numa estrutura de dados única, e então assumirem um significado para um contexto de aplicação mais específica. Isso significa ampliar o conceito de *features* implementado em Sistemas CAD comerciais. Nessa tese, informações de projeto serão denominadas de *features* de projeto. As *features* de projeto são consideradas como unidades de informação, servindo como meio de representação computacional para transmissão da informação de projeto de produto.

Pretende-se utilizar o conceito de modelagem por *features* para fazer a identificação de informações de cada uma das fases do projeto mecânico, buscando encontrar uma assinatura de cada fase do projeto. E dessa forma, o conjunto de *features* da fase atual deve ser uma evolução da informação geométrica e não-geométrica da fase anterior.

A abordagem inicial consistiu na identificação e classificação de *features* que ocorrem nas diferentes fases do processo de projeto mecânico. Para isso foram implementadas as classes de *features* próprias de cada fase do projeto, as quais editam a informação não-geométrica necessária para dar um significado de aplicação. Definidas essas classes de *features* das fases de

projeto, fica possível através de uma biblioteca de funções de um sistema CAD – API (*Application Programming Interface*) – a inserção das *features* para dentro do ambiente de modelagem de um sistema CAD convencional, a fim de criar o projeto detalhado do produto muito mais completo, com informações também de outras fases do processo de projeto.

Essa tese busca uma solução que facilite o projetista desenvolver sua atividade principal – que é projetar – mas contribuindo para o desenvolvimento de sistemas com interface computacional melhorada e integrada. É importante também que, para fins de sistematização e facilidade de integrar-se ao projeto e demais etapas subseqüentes do ciclo de vida do produto, propor que a modelagem de dados considere a evolução das informações de projeto com base numa sistemática de projeto de fases bem definidas.

A partir do sistema CAD, aliado a uma nova visão de aplicação dessa ferramenta no processo de projeto, é que se vislumbrará a integração necessária para utilização de outros sistemas e recursos computacionais já disponíveis, tais como: bases de dados de componentes padronizados, bibliotecas de *features*, banco de dados de funções de produtos e princípios de solução e regras especialistas. Maiores detalhes sobre os recursos computacionais disponíveis nos sistemas CAD são citados no Capítulo 2.

No desenvolvimento desses pontos de discussão, exige-se conhecimento adquirido e experimentado dos princípios de modelagem orientada a objeto e de tecnologia de *features*, além de conhecimento de como são definidos, representados, e processados as informações e os dados de projeto. O conhecimento prévio dessas técnicas de modelagem é básico para visualizar contribuições nesta área de projeto auxiliado por computador, e o que é mais importante – propor alternativas de melhorias para sistemas computacionais de suporte ao PPP. O Capítulo 3 traz mais detalhes sobre a discussão entre os conceitos de modelagem orientada a objetos e as definições sobre a tecnologia de *features*.

É importante deixar claro a posição de destaque do sistema CAD nessa tese, servindo como uma plataforma-cliente para a implementação de outros sistemas computacionais de suporte ao PPP.

1.4. OBJETIVOS

Utilizando os princípios de modelagem orientada a objeto e de *features* de projeto, pretende-se:

- Definir classes de *features* que encapsulem os atributos relevantes para capturar e caracterizar a evolução e transformação das informações e dados de projeto;
- Contextualizar os conceitos de *features* de projeto e objetos, ressaltando similaridades e diferenças;
- Proporcionar o reuso, a troca e o compartilhamento das informações de projeto através de uma base genérica de classes de *features* de projeto;

Os objetivos específicos citados acima, no contexto dessa tese, orientam o objetivo geral, a saber: “a modelagem de informações baseadas em objetos aplicados às fases do processo de projeto de produto”. Nessa perspectiva, a presente tese qualifica e seleciona dentre os modelos computacionais, o modelo de objetos como o que mais se aproxima e pode atender aos requisitos da atividade de projetar.

Juntos, os conceitos e modelos de objetos e *features* definidos nessa tese, possibilitarão a captura do significado de Engenharia embutido nas entidades gráficas de um sistema CAD convencional. Além disso, as *features* identificadas devem servir de informação útil para o projetista, e resultar em sistemas computacionais de suporte ao PPP mais integrado às fases de projeto informacional, conceitual, preliminar e detalhado.

Um sistema CAD associado às fases do processo de projeto através da modelagem de *features*, favorece a troca e o compartilhamento de informações do processo de projeto, de forma a possibilitar que o projeto seja melhor gerado, interpretado e organizado por um sistema computacional.

Capítulo 2

Processo de Projeto de Produto e Sistemas CAD

2.1. INTRODUÇÃO

O papel da Engenharia no desenvolvimento tecnológico é realizado pelos engenheiros que influenciam o cotidiano das pessoas de várias maneiras, tais como: construções e equipamentos; veículos e rodovias; o crescimento da safra de alimentos; máquinas de impressão de livros; produção de remédios; imagens de TV; etc..

A atividade econômica e a base do progresso tecnológico são resultados de uma união de esforços de Engenharia, Processos de Fabricação e Produção.

Atualmente, os engenheiros precisam de ferramentas e conhecimento de processos automatizados e computadorizados para auxiliá-los no trabalho. Os computadores atuam como processadores de dados. Eles assistem à definição e ao processamento de dados integrados ao projeto do produto, com a organização e o gerenciamento dos Sistemas de Fabricação.

Segundo (CHANG *et al.*, 1997), o sistema de fabricação é uma organização que compreende vários subconjuntos de fabricação inter-

relacionados, a saber: Projeto, Fabricação, Produção, Planejamento e Controle. Esses, por sua vez, inter-relacionam-se com as funções externas de produção: Marketing, Publicidade, Contabilidade, Economia e Finanças. O objetivo é otimizar o rendimento global do sistema, tais como tempo de produção, custo, e utilização de recursos produtivos (*homem, máquina, matéria-prima, etc.*). A Figura 1 ilustra os subconjuntos que constituem o sistema de fabricação e o seu relacionamento com as funções externas de produção.

O resultado gerado pelo sistema de fabricação pode ser classificado em duas categorias:

- O resultado físico, ou produto;
- A informação do rendimento da fabricação, que geralmente serve como um retorno ou ajuste para as máquinas ou mecanismos de controle.

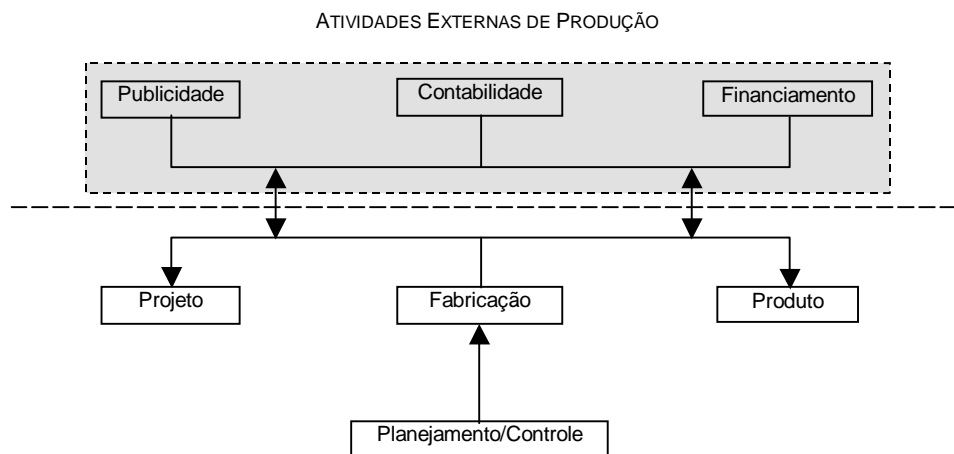


Figura 1. Sistema de fabricação e atividades externas de produção (Baseada em CHANG *et al.*, 1997).

Sendo assim, é importante enfatizar o papel dos sistemas computacionais na geração e no gerenciamento da informação de projeto que descreve o produto; e mostrar como computadores são usados no gerenciamento da informação sobre os sistemas de fabricação que produzem os produtos.

Portanto, pensar em modelagem de informações na atividade de projeto deve levar em conta a atividade de Engenharia, e os meios que um produto chega aos consumidores, processo de fabricação, uso e bens.

Para inserir os computadores no processo de projeto, é fundamental entender esse processo de projeto e as informações que são processadas.

2.2. PROCESSO DE PROJETO

Existem várias tentativas de fornecer uma descrição formal dos estágios ou elementos do processo de projeto (OGLIARI, 1999). Dessas tentativas surgem variantes, em termos de terminologia e detalhes, mas que no geral concordam nas seguintes atividades:

- Declarações de necessidades pela identificação do problema (*a especificação de requisitos*);
- Busca por soluções; e
- Desenvolvimento e implementação da solução escolhida para fabricar, testar e usar o produto.

As descrições detalhadas do processo de projeto constituem os modelos do processo de projeto. De todas as sistemáticas descritas na literatura científica, dois modelos de referência serão comentados, a saber:

- Sistemática de Pahl e Beitz, em 1977;
- Modelo de Oshuga, em 1989, segundo (OSHUGA, 1989).

Na sistemática de Pahl e Beitz, mostrada na Figura 2, o processo de projeto é descrito por um diagrama de fluxo constituído por quatro fases principais, resumidas aqui em:

- (1) **Definição da tarefa:** envolve a coleta de informações sobre os requisitos de projeto, e as restrições do projeto gerando o que se chama de “especificações de projeto”. Essa fase é denominada por (FONSECA, 2000), de **Projeto Informacional**;
- (2) **Projeto Conceitual:** envolve o estabelecimento de funções do produto, e a identificação e desenvolvimento de soluções alternativas para essas funções;
- (3) **Projeto de Encorpamento (*Embodiment*) ou Projeto Preliminar:** nesta fase, a solução escolhida no projeto conceitual é desenvolvida, no sentido de selecionar materiais, dimensionar componentes e

sub-sistemas, definir esquemas de montagem, selecionar processos, e resolver outros problemas;

- (4) **Projeto Detalhado:** nesta fase, as dimensões, tolerâncias, materiais e a forma dos componentes individuais de projeto são especificados em detalhes para as subseqüentes fases de fabricação e produção do produto.

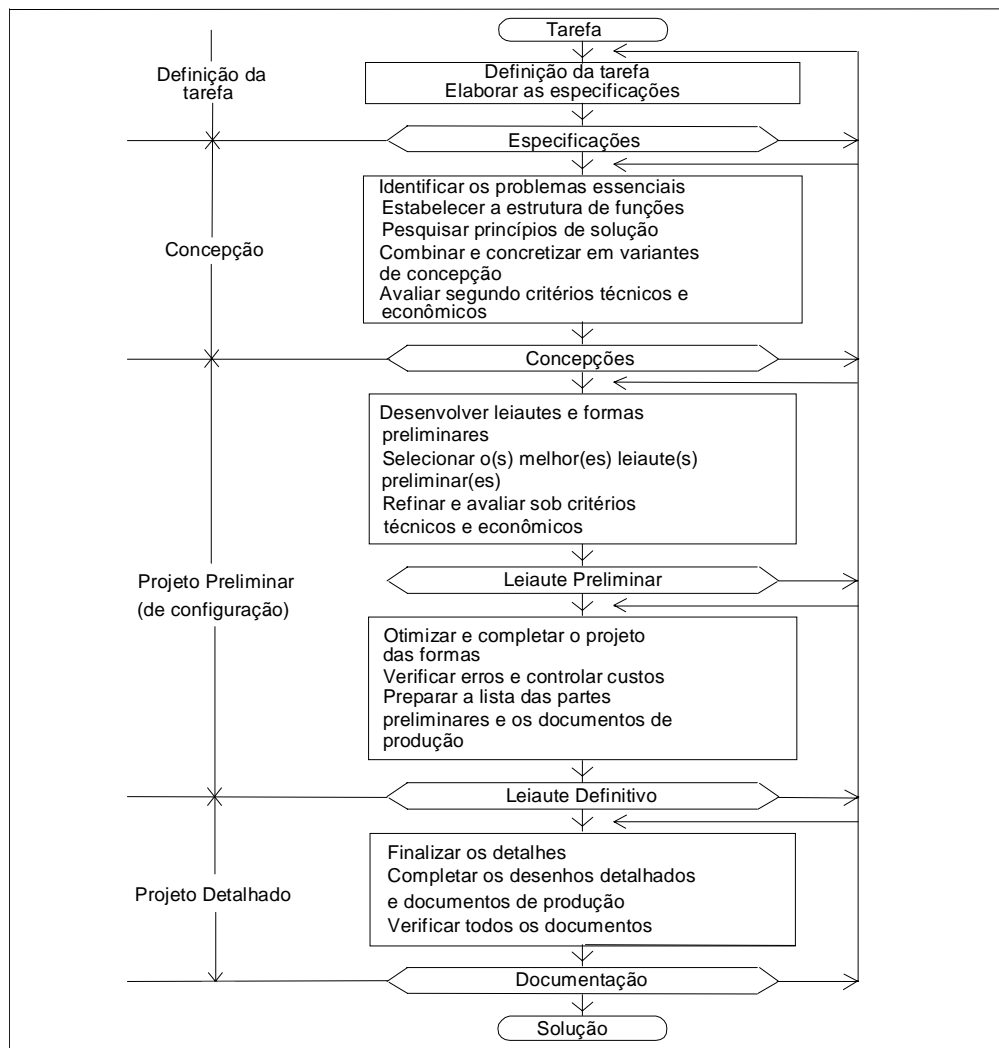


Figura 2. Sistemática do processo de projeto segundo Pahl e Beitz (Baseada em PAHL e BEITZ, 1996).

Embora a sistemática de Pahl e Beitz apresente uma seqüência de fases, na prática existem interações entre essas fases até um completo entendimento e desenvolvimento das soluções de projeto.

Esse ciclo de retro-alimentação do processo de projeto, através de interações e iterações entre as fases, é uma rotina no caso do Modelo de

Oshuga mostrado na Figura 3. Esse modelo descreve o processo de projeto como uma série de estágios, progredindo dos requisitos para o projeto conceitual, preliminar e detalhado. A característica é que nesse modelo, os vários estágios do processo de projeto são generalizados numa forma comum, a saber: os modelos do projeto passam por um processo de análise e avaliação para modificar e refinar o modelo ao longo das iterações. Cada iteração gera um modelo mais refinado e o leva para um estágio mais evoluído do processo.

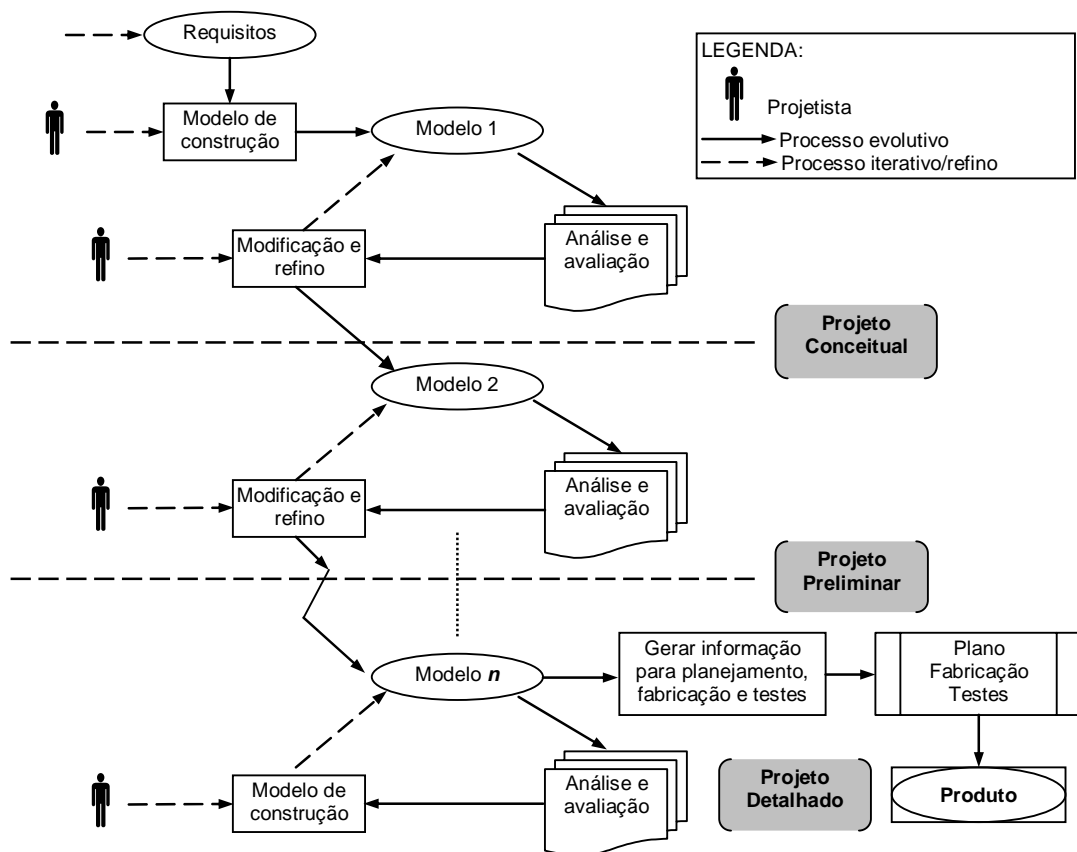


Figura 3. Modelo do processo de projeto segundo Oshuga (Baseado em OSHUGA, 1989).

Cada um dos modelos do processo de projeto, apresentados e discutidos acima, segue uma visão ligeiramente tradicional/clássica. Essa visão é caracterizada por uma seqüência de estágios de projeto, seguidos por processos de fabricação. Em suma são modelos seqüenciais.

Todavia, a busca pela redução de tempo no ciclo de desenvolvimento do produto está levando que companhias conduzam o projeto, desenvolvimento, análise e a preparação da informação de fabricação, em paralelo ou

simultaneamente. Essa nova visão é conhecida como Engenharia Simultânea ou Engenharia Concorrente (MCMAHON e BROWNE, 1998).

2.3. A INFORMAÇÃO NO PROJETO

Imagine-se um projetista, fazendo parte de uma equipe de projeto, onde não exista o emprego de nenhuma sistemática de projeto para desenvolver seus produtos. Geralmente essa forma de trabalho é marcada por certos vícios e paradigmas, frutos de alguma experiência adquirida em situações semelhantes anteriores. A discussão quanto ao emprego ou não de uma sistemática de projeto, pode tornar-se uma questão menor se os resultados finais já estiverem sendo alcançados satisfatoriamente. Vale ressaltar, que uma prática comum de procedimentos padronizados e sistematizados, adotada e aceita por outros, pode tornar muito mais eficiente a atividade de desenvolvimento de produtos.

Todavia, da forma descrita acima, o sucesso ou não das decisões de projeto só serão sentidos ao final se não forem providenciados meios de avaliação das decisões e resultados. E isso não é, nem muito bom nem muito saudável, do ponto de vista de mercado atual. Neste caso, o sucesso ou não das decisões tomadas é muito incerto.

A importância do emprego de sistemáticas de projeto vem sendo ressaltada por pesquisadores de projeto de sistemas mecânicos, pois elas tornam o processo de projeto muito mais orientado, gerenciável, e previsível, fornecendo parâmetros palpáveis e visíveis da eficiência ou não dos resultados, mesmo em fases ainda iniciais do processo. Por outro lado, alguns também ressaltam que a sistematização pode inibir ou limitar a criatividade. O ideal seria um meio termo, um pouco de flexibilidade e liberdade juntamente com um pouco de métodos e técnicas.

Essa discussão reflete que o desenvolvimento de um PPP é altamente imprevisível, aleatório, não determinista e dependente de fatores, que às vezes são alheios aos resultados pretendidos. Geralmente o resultado principal desejado pela equipe de projeto é uma solução para uma necessidade de um cliente. E esse quadro é até compreensível já que o processo de projeto é conduzido por seres humanos.

Dentro dessa dinâmica do processo de projeto, a informação é um dos elementos metodológicos de projeto. Nesta tese, deseja-se empregar uma metodologia de projeto como base para a modelagem da informação de projeto e posterior visualização de uma ferramenta computacional que se integre com visão metodológica prescrita. A abordagem deste trabalho, de início focaliza a discussão das formas de representação da informação de projeto e na sua transformação durante as fases.

A informação assume o papel de matéria-prima do processo de projeto, e dos resultados obtidos durante o projeto. Então, uma avaliação sobre as formas de representação da informação é de fundamental importância para o modelo de informação do projeto.

A principal preocupação com a forma de representação da informação de projeto não está na fase de detalhamento do projeto, onde as informações já assumem características bem definidas, e de certa forma já invariáveis. A preocupação maior está em definir meios computacionais de capturar a intenção do projetista nas fases iniciais de projeto, principalmente nas fases informacional e conceitual. E depois, utilizar a informação inicial para relacioná-la de forma inteligente, útil, e reutilizável do ponto de vista de projeto, com as formas de representação detalhada, já disponíveis nos sistemas CAD. Esse relacionamento entre ferramentas já disponíveis no sistema CAD deve priorizar, a integração para suportar o projeto. Isso exige um modelo computacional complexo, pois a representação da informação é diversa.

A intenção desse tópico é mostrar a representação da informação de projeto utilizada em trabalhos desenvolvidos na área de projeto mecânico de produtos industriais, de forma a exemplificar e reutilizar alguns esforços de pesquisa nesta área. O estudo feito aqui se baseou em teses desenvolvidas no âmbito das fases informacional (*especificação de projeto*) e conceitual (*geração e seleção de uma alternativa de solução conceitual*).

O projeto informacional é executado para transformar os dados de entrada em especificações de projeto. Estas especificações se constituirão no guia dos trabalhos nas fases posteriores do processo de projeto, razão pela qual a sua obtenção implica numa responsabilidade para o sucesso do projeto no seu conjunto. Na visão do desenvolvimento do produto no

contexto da fase informacional, a atividade de projetar um produto, pode ser representada pela Figura 4. A ênfase recai sobre o papel do documento de especificação de projeto, o qual se caracteriza como o principal documento que dar entrada no desenvolvimento do produto.



Figura 4. Visão do desenvolvimento do produto durante o projeto informacional.

Os pesquisadores estabelecem que o processo de projeto se inicia com o esclarecimento ou definição da “*tarefa de projeto*”. A atividade de projeto deve ser precedida por um trabalho da equipe de “*marketing*”, executado para definir, dentre o universo de produtos possíveis, aquele produto que vai ser projetado e configurar, assim, a “*tarefa de projeto*”, como o documento que dá início ao processo de projeto.

A Figura 5 mostra a seqüência de processamento das informações de projeto e as respectivas transformações dos dados no projeto informacional.

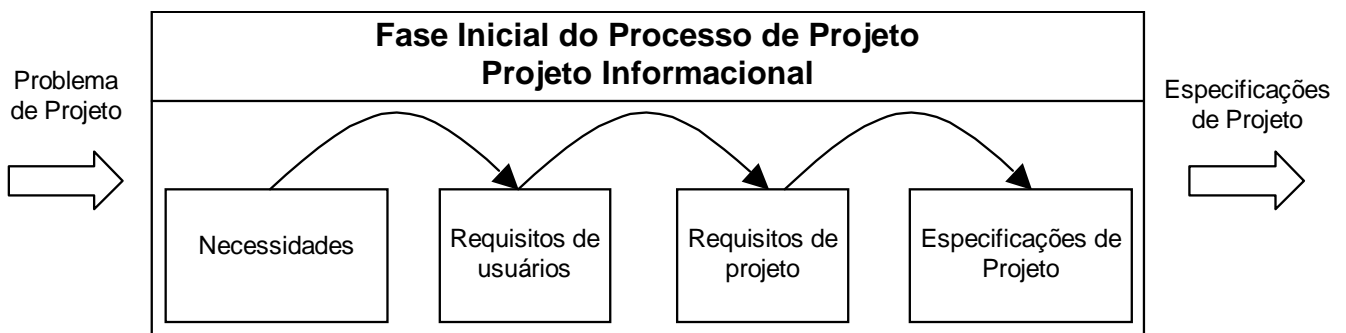


Figura 5. Seqüência de processamento das informações de projeto e transformação dos dados no projeto informacional (FONSECA, 2000).

Para uma adequada estruturação do projeto informacional, consideram-se quatro categorias de informação, como categorias relevantes realmente existentes no processo: as necessidades, os requisitos de usuário, os requisitos de projeto e as especificações de projeto. Elas estão resumidas na Tabela 1.

Tabela 1. Categorias de informação na fase de projeto informacional (FONSECA, 2000).

TIPO DE INFORMAÇÃO	SIGNIFICADO
Necessidade	Declaração direta de usuário ou clientes
Requisito de usuário	Necessidade, levada à linguagem de projeto
Requisito de projeto	Requisito mensurável, aceito para o projeto
Especificação de projeto	Característica de projeto e/ou do produto

Na realidade, estas informações mínimas devem estar contidas no problema de projeto, porém, deve-se revisar o problema visando complementá-las. Conforme recomendações de (FONSECA, 2000), os dados a serem levantados antes de iniciar o trabalho são:

- Dados do estudo de *marketing* prévio (*revisão do documento*);
- Tipo de produto;
- Tipo de projeto;
- Volume planejado de fabricação;
- Desejos explícitos expostos no problema de projeto; e
- Restrições do projeto ou do produto.

Esses dados devem estar registrados num documento chamado de “ordem de projeto”, o qual deve conter o mencionado problema, seja procedente da equipe de *marketing*, ou procedente do ambiente externo. Na ordem de projeto referida devem aparecer, obrigatoriamente, os seguintes dados: objetivos; metas; restrições; desejos explícitos e descrição do problema de projeto.

Na fase inicial tem-se a necessidade de desenvolver as especificações de projeto. Esse processo é auxiliado por ferramentas ou matrizes.

A matriz das necessidades, como mostrada na Figura 6 é constituída por linhas representando as fases do ciclo de vida do produto, e as colunas representadas pelos atributos básicos do produto. No cruzamento de linhas e colunas busca-se o estabelecimento de necessidades de projeto na forma de requisitos de usuário.

Atributos básicos do produto						
Ciclo de Vida	Funcionamento	Ergonomia	Estética	Econômico	Normalização	Modular
Produção		Ter fácil soldagem				
Montagem		Ter facilitada a montagem				
Transporte				Ter facilidade de transporte		
Armazenagem				Ter facilidade de armazenagem		
Função	Ter porta-material; Ter mesa c/ porta-material		Ter cor agradável	Ter estrutura leve		Estrutura modular e resistente
Uso	Ter mesa mais larga; Ter mesa inclinada; Ter encosto maior	Ser ergonômica, Não seja dura, Não ter ressaltos				
Manutenção				Ter facilidade de manutenção	Ter uniões normalizadas	

Ser pintada sem desperdício

Ter mínimo tempo de produção

Ter custo mínimo de produção

Figura 6. Matriz de levantamento de requisitos de usuário (FONSECA, 2000).

Na Figura 7 os requisitos de usuário aparecem como linhas da matriz, tendo como colunas os atributos específicos do produto.

Atributos específicos do produto											
Requisitos de usuário	Geométricos	Material	Cor	Peso ou massa	Forças	Cinemática	Energia	Fluxo	Sinais	Estabilidade	Qualidade
Ter fácil soldagem	Reduzir juntas complexas										
Ser pintada sem desperdícios	Usar peças similares										
Ter mínimo tempo de produção	Elementos normalizados										
Ter mínimo custo de produção	Mínimo de peças										
Ter facilitada a montagem	Reduzir juntas complexas										
Ter facilitado o transporte	Formas encaixáveis										
Ter facilitada a armazenagem	Formas empilháveis										
Ter porta-material na cadeira e na mesa	Usar a estrutura para o porta-material										
Ter cor agradável		Madeira e tubo de aço	Evitar cores vivas								
Ter estrutura leve	Estrutura modular simples										
Ter estrutura modular e resistente	Decidir seções dos tubos										
Ter mesa e encosto maiores	Incrementar as áreas de mesa e encosto										

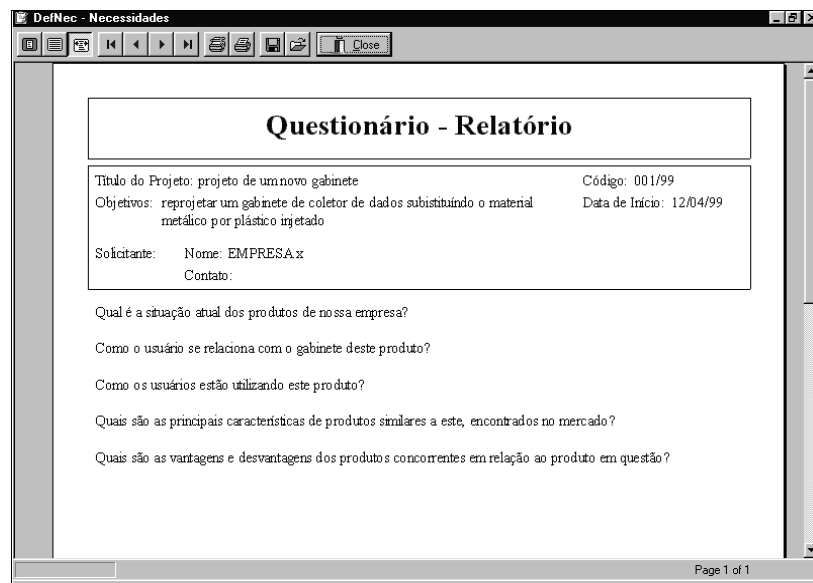
Figura 7. Matriz para conversão dos requisitos de usuário em requisitos de projeto (FONSECA, 2000).

Nessa matriz da Figura 7, o cruzamento de uma linha (*requisito de usuário*) com uma coluna (*atributo específico*), identifica e ajuda a equipe de projeto a decidir qual o requisito de projeto, que é algo mensurável, satisfaz o requisito de usuário da linha correspondente.

Outra forma de capturar a informação das necessidades e requisitos do usuário nas fases informacional e conceitual é através de questionário

estruturado e listas de verificação (*checklist*) relacionados ao domínio do produto, conforme está exemplificado pela Figura 8.

Numa etapa seguinte do processo de projeto – o projeto conceitual – desenvolvem-se e representam-se as funções do produto, com vistas a estabelecer o problema de projeto na forma funcional. Conforme afirma (OGLIARI, 1999), esse é um processo de faturação ou “Divida e Conquiste” (*Divide & Conquest*).



The screenshot shows a window titled "DefNec - Necessidades" with a toolbar at the top. The main content is a form titled "Questionário - Relatório". The form contains the following text:

Título do Projeto: projeto de um novo gabinete Código: 001/99
Objetivos: reprojeter um gabinete de coletor de dados substituindo o material metálico por plástico injetado Data de Início: 12/04/99

Solicitante: Nome: EMPRESA.x
Contato:

Qual é a situação atual dos produtos de nossa empresa?
Como o usuário se relaciona com o gabinete deste produto?
Como os usuários estão utilizando este produto?
Quais são as principais características de produtos similares a este, encontrados no mercado?
Quais são as vantagens e desvantagens dos produtos concorrentes em relação ao produto em questão?

Page 1 of 1

Figura 8. Exemplo de questionário estruturado para levantamento de necessidades (OGLIARI, 1999).

As funções do produto podem ser representadas por ícones ou símbolos indicativos ao nome da função. Esses ícones podem ser relacionados através de uma árvore hierárquica, de forma a compor a estrutura de funções do produto. Na Figura 9 tem-se um exemplo desse tipo de informação. A estrutura de funções do produto estabelece uma solução funcional particularizada.

O modelo seguinte, utilizado para representar os princípios de solução que atendem às funções e permitem comparar e avaliar diferentes alternativas de solução é desenvolvido com auxílio da matriz morfológica. Conforme mostrado na Figura 10, tem-se os elementos do domínio do problema de projeto e as funções do produto nas duas primeiras colunas. As demais colunas da matriz são preenchidas com princípios de solução que compõem alternativas de solução para o problema.

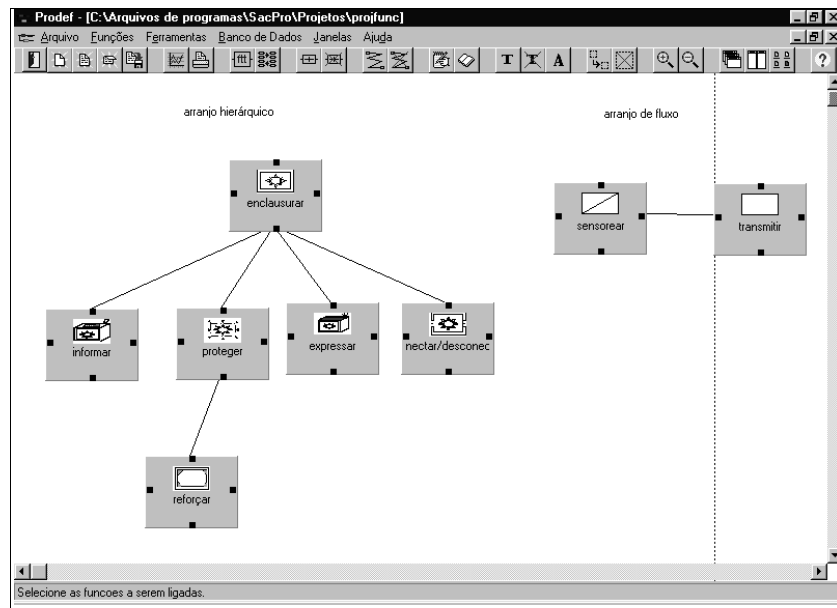


Figura 9. Estrutura de função do produto numa forma hierárquica e representada por ícones (OGLIARI, 1999).

Elementos do domínio	Funções do gabinete	Concepções alternativas		
		Concepção 1	Concepção 2	Concepção n
Componentes internos	enclausurar componentes internos 			
	fixar componentes internos 			
Usuários do produto	etc.			
	informar o usuário do produto 	TEXTO 	FORMA 	TEXTO
Ambiente do produto	vedar contra influências nocivas do ambiente 			
	etc.			
Demais sistema técnicos	combinar com outros produtos 			
	etc.			
Funções especiais	função processo			
	função molde			
	função material			

Figura 10. Exemplo de matriz morfológica para representação das alternativas de solução do problema (OGLIARI, 1999).

A partir das formas de representação utilizadas para representar as informações de projeto nas fases iniciais, conclui-se que:

- Para o relacionamento de informações nas fases informacional e conceitual, a forma mais usual de visualização de grupos informações de projeto é feita através de tabelas ou matrizes. Essas

tabelas, normalmente podem ou devem ser vinculadas a banco de dados (DBMS – *Database Management Systems*), os quais proporcionam aos usuários das ferramentas de suporte ao projeto todas as vantagens de um banco de dados (CUNHA e DIAS, 2000a).

Nesta discussão sobre a caracterização da informação de projeto visando capturar a evolução dos dados de projeto, é importante ter um entendimento detalhado dos conceitos e relacionamentos entre *features*, objetos, restrições e decisões no contexto do processo de projeto, os quais constituirão a informação de projeto desta proposta, e que nortearão a modelagem das classes de *features*.

2.4. SISTEMAS CAD E O PROCESSO DE PROJETO

No começo da utilização dos sistemas CAD nas atividades de projeto, a principal preocupação se concentrou na automação das atividades relacionadas à geração de desenhos técnicos. Os sistemas CAD eram vistos e usados como pranchetas eletrônicas estilizadas.

Na maioria dos casos, a aplicação deles se resumia a automatizar atividades com características tais como:

- Atividades repetitivas e tediosas: geralmente esse tipo de atividade não acrescenta nenhum ou pouco valor técnico de Engenharia à atividade principal, que no caso do CAD, deveria ser de projetar e de suportar um processo de projeto;
- Atividades que possuem uma seqüência de implementação definida e padronizada; ou seja, adequada a um processo de automação.

O objetivo principal dos Sistemas CAD era aumentar a produtividade e a documentação do desenho detalhado do produto. Esse objetivo foi alcançado, de certa forma, quando os recursos do CAD eram aplicados correta e extensivamente no ambiente de trabalho. Todavia, é fato que a ferramenta adequada não é condição suficiente para garantir bons resultados. Ela apenas habilita o usuário a conseguir o objetivo pretendido.

A adaptação ao uso de uma nova tecnologia ou à mudança de procedimentos, mesmo que através de uma boa ferramenta, sempre exige

um esforço inicial e alguns percalços até se conseguir o domínio pleno ou satisfatório da ferramenta. Esse período de adaptação depende de alguns fatores, sendo que os principais e mais decisivos apontados aqui são:

- **Nível de conhecimento teórico/prático do usuário:** é desnecessário, senão redundante enfatizar que experiências anteriores com a utilização, ou um conhecimento teórico embasado no potencial de um sistema CAD genérico, ou ambos, são necessários para utilização satisfatória dos recursos de produtividade oferecidos por um sistema CAD. O mais importante e que exige mais perícia, atenção e tempo, é saber visualizar as limitações na aplicação do sistema.
- **Facilidade que a ferramenta tem de oferecer, quanto ao uso, os seus recursos computacionais:** neste ponto é decisiva a importância da interface do usuário. Usar uma interface mal projetada induz erros ou deixa o usuário indeciso durante a realização de algumas operações, ou pode até mesmo embutir vícios na forma de trabalhar de um usuário mais incauto ou inexperiente. No sistema CAD, uma interface amigável deve conduzir o usuário na realização da operação, de forma a levá-lo ao término dela, sem atropelos e sem a necessidade de tentativas prévias para entender ou dominar o recurso.

2.4.1. MUDANÇAS NA APLICAÇÃO DO SISTEMA CAD

O projeto de Sistema CAD e suas aplicações sempre foram pensados da mesma forma que de um sistema computacional qualquer, isto é, que se assemelha a outros sistemas computacionais no que concerne à interface, recursos e funcionalidades de interação com o usuário. Esse quadro situacional reflete o pensamento da época da introdução dos computadores dentro da rotina do ambiente de trabalho, onde se buscava apenas automatizar tarefas (BARBOSA *et al.*, 2003).

Tem-se exigido dos sistemas CAD uma adaptação à realidade de um mercado consumidor ansioso por diversidade e qualidade nos produtos fabricados, e ao dinamismo da economia globalizada. Na rotina das

empresas e escritórios de projeto, isso é refletido pelo aumento do fluxo de informações e dados inter-setores, os quais precisam estar integrados dentro do processo de projeto e do fluxo de trabalho. Num ambiente como este, a medida de produtividade do sistema CAD está associada diretamente à flexibilidade e aos seus recursos computacionais. São essas qualidades que habilitam o sistema CAD se integrar às atividades de projeto e desenvolvimento do produto. Essa não é só uma exigência feita aos sistemas CAD, mas a todas as ferramentas computacionais de suporte ao projeto, de que a documentação dos dados e informações relacionadas ao projeto seja de alguma forma armazenada e capturada adequadamente no computador.

Atualmente, os sistemas computacionais de suporte ao processo de projeto mecânico buscam modelar e capturar todas as informações do ciclo de vida do produto. Essa é uma tendência observada por muitos pesquisadores relacionados à área, e conforme é reforçada em (MYUNG e HAN, 2001). A intenção desses sistemas é estabelecer e definir um modelo do produto completo de informação associada a contextos específicos, desde a informação puramente geométrica como também, e principalmente, as informações não geométricas, sejam elas implícitas (*funções do produto, intenção de projeto, significado de Engenharia, etc.*) ou mais explícitas (*requisitos de projeto, custo, material, etc.*).

No caso dos sistemas CAD, a técnica mais atual para lidar com essa proposta de modelo do produto integrado ao ciclo de vida é a tecnologia de *features*. *Features* são o resultado de uma evolução da modelagem da informação adotada nos modelos geométricos em sistemas CAD. A Figura 11, baseada no artigo de (SHAH e ROGERS, 1988), mostra que a partir dos modeladores de estrutura de arame (*wireframe*) até os modeladores sólidos, a preocupação era simplesmente em capturar aspectos puramente geométricos do produto. Com os modeladores baseados em *features*, passou-se a modelar também aspectos não geométricos associados à definição completa do produto. Tais aspectos são modelados considerando três fatores principais: o nível de abstração da informação ou dado, o tipo de aplicação e o domínio do produto.

É possível afirmar que a tecnologia de *features* trouxe uma mudança de paradigma na aplicação dos sistemas CAD dentro do processo de projeto. A proposta moderna e atual do conceito de *feature* direciona os sistemas CAD focados na informação, onde não só apenas o conhecimento, mas a informação correta, atualizada, contextualizada e organizada fazem a diferença na obtenção de produtos de melhor qualidade.

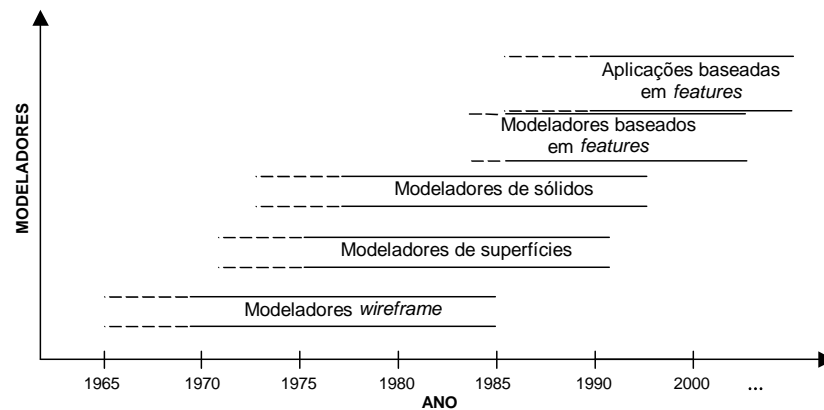


Figura 11. Evolução da modelagem geométrica dos modelos em sistemas CAD.

Devido à importância dos sistemas CAD e CAX's em geral, justifica-se a procura dos aspectos necessários para otimização da aplicação de tais ferramentas no suporte ao projeto. Os sistemas CAD exercem um papel importante de documentação, gerenciamento, compartilhamento e troca de dados de projeto na maior parte das atividades de um sistema de fabricação.

As ferramentas de projeto baseadas no computador são amplamente disponíveis para as últimas duas fases, respectivamente as fases de Projeto de Preliminar (*encorpamento – embodiment*) e Projeto Detalhado. Tais ferramentas resumem-se a dois tipos: modeladores de sólido e sistemas CAD, os quais modernamente estão baseados em tecnologia de *features*.

(ULLMAN *et al.*, 1990) adverte que o suporte proporcionado pelos sistemas CAD concentra-se muito mais em suportar o desenho (“**D**” – **DRAFTING**) em si, do que atender às necessidades de projeto (“**D**” – **DESIGN**).

Wigard, citado por (MAZIERO, 1998), afirma que nos programas CAD/CAM atuais, o usuário deve expressar o conceito que ele quer representar usando elementos relacionados à representação interna do modelo. Isso pode mascarar o real significado dos conceitos, e levar o

raciocínio e a criatividade do projetista para questões não relacionadas com a sua atividade principal, que é projetar. E essa forma de visualizar a ferramenta CAD está passando por um processo de grandes mudanças. Conforme já advertido, o CAD deve deixar de ser apenas desenho (*drafting, drawing*) e assumir a sua verdadeira identidade, o projeto (*design*).

Através de experimentos e uma análise dos resultados obtidos, (ULLMAN *et al.*, 1990) também assinalou quatro áreas onde a ferramenta CAD pode e deve se desenvolver. São questões onde ainda muitas pesquisas podem ser realizadas. Essas estão resumidas e comentadas abaixo:

- Permitir e oferecer entradas de informações e dados de projeto por meio de rascunhos, uma forma rápida e comumente usada por engenheiros e projetistas para materializar idéias. E essa informação tem os seus méritos e seu valor técnico;
- Permitir uma variedade de interfaces para o projetista. Isto não significa disponibilizar mais formas para definir um círculo, mas selecionar e padronizar uma interface que aproxime o usuário da ferramenta computacional;
- Reconhecer *features* dependentes do contexto de aplicação e tratá-las como objetos;
- Ser hábil a gerenciar restrições abstratas e restrições funcionais, e garantir ou sugerir solução para elas.

Embora a modelagem geométrica represente um importante avanço e melhoramento sobre os sistemas CAD, estes ainda possuem algumas deficiências que precisam ser consideradas para efetivamente servirem como meio de integração com outras atividades relacionadas à Engenharia. Abaixo, listam-se algumas dessas dificuldades ou questões ainda não otimizadas pelos sistemas CAD (SHAH e MÄNTYLÄ, 1995) :

- Interpretação da informação geométrica a partir do ponto de vista de fabricação;
- Disponibilidade de informação associada – tipo não-geométrica – necessária para um processo subsequente de planejamento;
- Dificuldade de integrar sistemas especialistas ou modelos de análise com modelos geométricos;

- Falta de padronização na forma de representação computacional da informação e do dado de projeto no CAD;
- Dificuldade de processamento de dados abstratos, que muitas das vezes dizem respeito à intenção do projetista;
- As bases ou modelos de dados de informação não suportam todas as informações processadas no ciclo de desenvolvimento do produto.

Uma primeira consequência é que a prática de projeto atual baseada no computador freqüentemente carece de uma sistemática que reflita as práticas ditadas pelas metodologias de projeto da teoria (CUNHA e DIAS, 2000b) (VAN VLIET *et al.*, 2001).

Uma segunda consequência torna-se aparente, é tentado adicionar inteligência ao CAD.

O atual status dos sistemas CAD o qualifica como um manipulador avançado de informações geométricas.

Assim, pode-se concluir que:

- Regras de projeto e procedimentos de análise são limitados unicamente à geometria.
- É exigido desenvolver ferramentas para os estágios iniciais do processo de projeto, onde a geometria pode ainda não está definida.
- É necessário também integrar as ferramentas de projeto baseadas no computador com sistemas CAD.

2.4.2. RECURSOS COMPUTACIONAIS DOS SISTEMAS CAD

Certas características e recursos computacionais de um sistema CAD são, geralmente, mais requeridos e necessários do que outros. De acordo com (WAKEFORD e FAY, 1998), quando usuários procuram por um novo sistema CAD para atender às suas necessidades mais básicas e comuns, eles listam os seguintes itens:

- Habilidade de importar dados a partir de outros sistemas CAD;
- Integração e conectividade com sistemas CAM;
- Interface do usuário.

Essas características e recursos do sistema CAD acabam servindo de critério decisivo para escolha, pois elas servem de apoio básico para utilizá-lo eficientemente e produtivamente. Os dois primeiros itens listados acima estão diretamente relacionados ao formato de arquivo proprietário do sistema CAD, e à estrutura de dados e informações.

Os sistemas CAD oferecem algumas ferramentas e recursos de personalização, os quais permitem desenvolver e implementar outros recursos estendidos e mais ajustados às necessidades específicas de um usuário. A título de exemplo, citam-se alguns recursos encontrados no sistema CAD MicroStation/J Modeler, da *Bentley*.

- Modelagem paramétrica: são recursos e funcionalidades de modelagem de formas (Perfis bidimensionais – *Sketches/Profiles*), os quais possibilitam a criação de relacionamentos geométricos e dimensionais, que por sua vez permitem fazer alterações de forma e reuso da informação gráfica a partir dos parâmetros definidos. Exemplo na Figura 12;

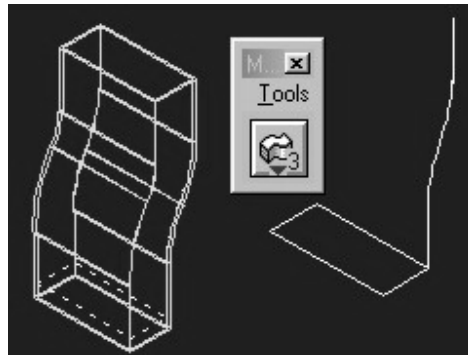


Figura 12. Exemplo de perfis bidimensionais gerando formas curvas e sinuosas.

- Desenvolvimento de bibliotecas de células personalizadas;
- Definições de *features* básicas;
- Bibliotecas de componentes padronizados: Figura 13 e na Figura 14;

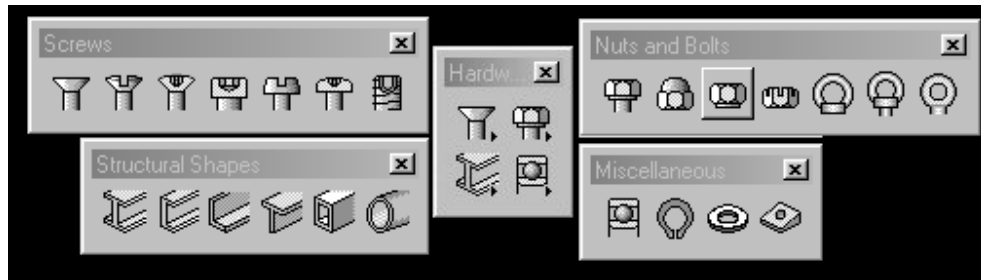


Figura 13. Biblioteca de componentes padronizados do MicroStation/J Modeler v7.1.

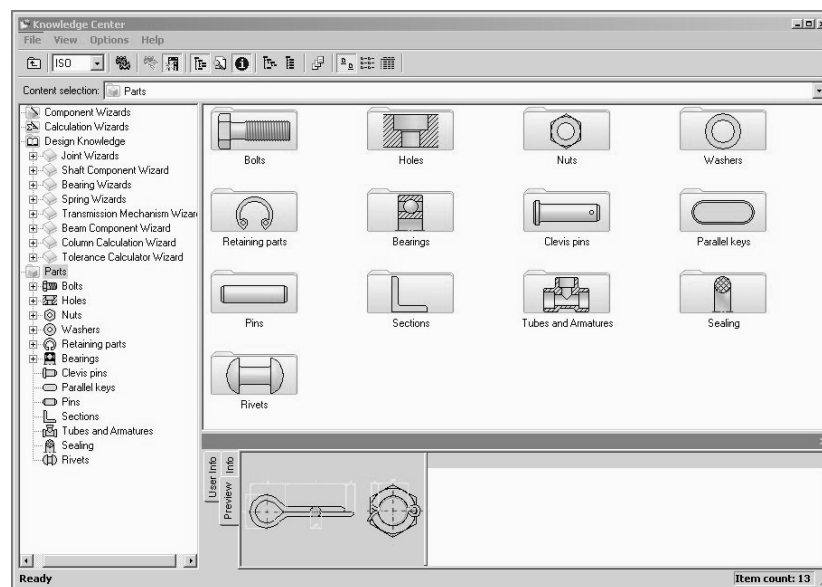


Figura 14. Biblioteca de componentes padronizados da MechSoft, Inc.

- Interface de integração com sistemas de banco de dados, permitindo o sistema CAD ler informações a partir de um banco de dados;
Ex.: Através de *drivers* ODBC – *Open Database Connectivity*, da *Microsoft*; ou JDBC – *Java Database Connectivity*, da *Sun Microsystems*.
- Interface de integração com linguagens de programação, através de API's próprias do sistema CAD.
Ex.: Basic; MDL (*MicroStation Developer Language*) em C; e JMDL, seu correspondente, em Java.

À medida que as informações e os dados vão se tornando mais complexos, o auxílio de ferramentas computacionais ao projeto torna-se indispensável na realização de tarefas mais criativas, de gerenciamento e suporte eficiente da dinâmica de projeto de um produto, como por exemplo:

- **Aumento da eficiência dos métodos de modelagem do produto:** captura da intenção do projetista dentro do modelo geométrico gerado pelo sistema CAD; geração e seleção automática de soluções alternativas a partir de uma especificação textual ou gráfica de projeto;
- **Gerenciamento e manutenção das informações e dados de projeto:** estrutura de dados representativa do contexto de projeto, reusáveis e extensíveis em aplicações subseqüentes do ciclo de vida do produto;
- **Integração com o processo de projeto e fluxo de trabalho da empresa:** interoperabilidade com outras ferramentas computacionais; interfaces computacionais personalizadas e ajustadas para aplicações específicas.

Dessa explanação de sistemas CAD conclui-se que a utilização dos recursos já disponíveis na maioria dos sistemas CAD classificados como de média e alta aplicação (*middle/high-end application*), habilitam o analista a propor recursos computacionais estendidos, que atendam e suportem operações comuns desejadas por um usuário do domínio de projeto. E a razão para esse direcionamento, é desenvolver recursos que facilitem para o usuário do sistema a captura da intenção do projetista na criação do modelo sólido.

Todavia, uma outra conclusão, é que devido às particularidades próprias da atividade de projetar, fica complicado disponibilizar um conjunto de ferramentas integradas que satisfaçam por completo as diferentes necessidades do usuário.

Por isso, a estratégia de abordagem do problema de modelagem tem o objetivo de visualizar as informações mais importantes das fases de projeto e os seus relacionamentos. E assim, poder concentrar esforços para implementar recursos computacionais direcionados para o processamento dessas informações mais relevantes.

Capítulo 3

Objetos e Features de Projeto

3.1. INTRODUÇÃO

O conceito atual de *feature* é resultado de uma constante evolução e contínua mudança. Ele se ampliou, incorporando e integrando-se ao ciclo de vida do produto como um todo, e não mais apenas restrito ao domínio da Fabricação ou do Planejamento do Processo. Orientado por essa abrangência, a pesquisa de tese aprofundou-se nessa discussão. Avanços e revelações sobre os resultados obtidos estão descritos e comentados neste capítulo.

As contribuições dessa investigação, quando fundamentada em conceitos básicos e genéricos, combinam perfeitamente nos objetivos citados numa definição do que é pesquisa (JORNAL DA CIÊNCIA, 2001):

*“Uma **pesquisa** é um processo orientado para expandir as fronteiras do conhecimento. Ela representa uma investigação ordenada e original que é coerente com uma linha de pensamento conceitual e teórica; persegue em sua intenção de mostrar evidências através de um método racional de ação e experimentação e tem sempre a intenção de descobrir novas informações ou desenvolver novos processos de transformação para viabilizar produtos ou serviços de informação.”*

As declarações descritas, comentadas e defendidas neste capítulo têm o objetivo de descrever o estado da arte em *feature* e nas tecnologias relacionadas ao conceito do que é *feature*.

3.2. CONCEITOS E DEFINIÇÕES

A terminologia empregada na área de processo de projeto é diversa, por vezes ambígua, dúbia ou mesmo conflitante no que diz respeito à padronização dos termos empregados para se definir, qualificar ou expressar idéias, conceitos e intenções de projeto. Ou seja, a terminologia dos conceitos que são utilizados não está totalmente definida ou padronizada, nem no campo teórico e muito menos na modelagem computacional do problema (SRINIVASAN, 1999). Isso se justifica por alguns fatores inerentes ao processo de formação do conhecimento individual do projetista sobre determinado assunto, o qual no que se refere ao processo de projeto é ainda muito incipiente, para não dizer inexistente.

Percebe-se na literatura um esforço inicial e informal, em tratar esse problema da padronização de termos técnicos. Essa não padronização da terminologia acaba tornando-se um empecilho a mais para a estruturação de uma implementação computacional consistente e abrangente no processo de projeto.

Por isso, dentro da idéia de montar uma base consistente de conceitos baseados na modelagem orientada a objeto e na tecnologia de *feature*, esta pesquisa se propõe a oferecer uma forma de modelar informações de *features* que capturam os conceitos no contexto de cada fase do processo de projeto.

Para tanto, é de suma importância analisar, discutir e comparar como são interpretados alguns conceitos e terminologias comumente usadas na linguagem de Desenvolvimento do Produto e Metodologia de Projeto.

3.2.1. INFORMAÇÃO E DADO

Baseando-se inicialmente num trecho extraído do (JORNAL DA CIÊNCIA, 2001), intitulado “*O que é Informação e o que é Ciência da Informação*” – nas palavras de Aldo de Albuquerque Barreto, que escreve na Lista de Divulgação da **ANCIB** (**A**ssociação **N**acional de **C**iência da

Informação) – a título de promover a discussão, tem-se uma primeira idéia de dois conceitos básicos relacionados a esse trabalho:

O que é informação? E o que é dado?

Informação: estruturas significantes com o objetivo de gerar conhecimento no indivíduo e no seu espaço.

E para acrescentar e consolidar mais um pouco a discussão, tem-se também que a ciência que trata com esse conceito e tenta sistematizá-lo – a **Ciência da Informação** – preocupa-se com os princípios e práticas da produção (*geração*), organização e distribuição da informação. Assim como, com o estudo dos fluxos da informação desde a criação até a sua utilização, e a sua transmissão ao receptor numa variedade de formas, através de uma variedade de canais.

Informação é definida nesse trabalho como um conjunto de dados, onde os relacionamentos e as regras associadas para uma aplicação estão bem definidos e associados a um contexto de aplicação. Já os dados são atributos da informação, os quais estão relacionados exclusivamente aos valores, e por isso eles não precisam de um contexto de aplicação. A informação é o resultado ou a saída do processo de transformação baseado em dados, os quais são as entradas. A informação depende da adição de conhecimento aos dados, a qual é feita através de relações e regras (*restrições*) do domínio de aplicação, tal como mostrado pela Figura 15. Concluindo, pode-se dizer que a informação é um dado com um contexto de aplicação.

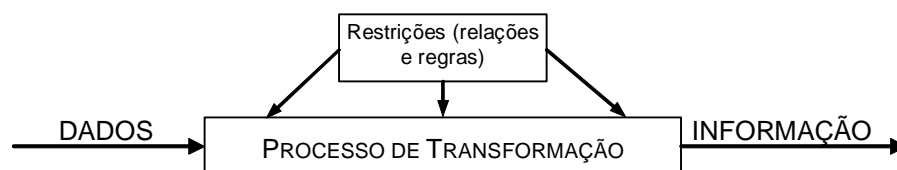


Figura 15. O processo de transformação de dados em informação.

Quando aplicado à atividade de projeto, a informação no processo de projeto vem associada aos substantivos usados para defini-la ou nomeá-la, e conseqüentemente possui uma semântica relacionada a ela. Fazendo-se uma correspondência com o processo de identificação de classes no contexto de análise de orientação a objeto, a informação de projeto encaixa-se perfeitamente com o conceito de classe de objetos. E por sua vez, os dados

de projeto referem-se a valores, quantidades, dimensões, etc.. Eles são fatos mais reais (*menos abstratos*), exatos (*sem dubiedade de interpretação*) e pontuais (*referem-se a instâncias*). Tudo isso os colocam muito mais próximos da idéia de instância de objeto.

Outra diferença entre informação de projeto e dado de projeto está na representação utilizada por ambos. Os dados de projeto podem ter várias formas para representá-lo, enquanto a representação da informação de projeto, uma vez definida, mantém-se a mesma em toda a existência do contexto ao qual ela se aplica. Isso implica que a informação de projeto possui, e deve ser assim, uma natureza muito mais invariável. Uma informação de projeto específica/particular não deve mudar durante o processo de projeto, porque ela diz respeito à semântica, a qual deve ser fixa durante todo o processo. Por sua vez, os dados de projeto transformam-se e modificam-se, visto que eles armazenam os valores geométricos e/ou não-geométricos da informação de projeto. Para exemplificar essa discussão, considere a informação de projeto de um furo passante numa peça. Se não houver um controle dinâmico da semântica “ser passante”, pode-se alterar a geometria do furo – mais especificamente sua profundidade – e assim obter um furo não-passante, sendo que o seu atributo “tipo” pode continuar com valor passante. Esse exemplo caracteriza uma situação de invalidação de semântica, onde a geometria não corresponde à semântica real.

A Figura 16 mostra através do conceito de estado de transição, que mudanças e transformações ocorrem durante as fases de projeto. E isso se dá quando o projetista especifica as restrições e toma decisões de projeto.

De acordo com o ponto de vista de (MCGINNIS e ULLMAN, 1992), o papel da informação nas atividades de projeto é dar um significado ou uma semântica ao dado de projeto. Para projetar um componente, algumas restrições sobre sua forma e sua função são fornecidas ainda nos estágios iniciais das fases de projeto. Algumas restrições são oriundas do conhecimento e da experiência do projetista sobre o domínio do problema, e outras são derivadas de cada decisão de projeto tomada no processo de projeto. (MCGINNIS e ULLMAN, 1992) concluíram que o refinamento do projeto de um componente depende de um conjunto de inter-relacionamentos entre as restrições de projeto. Eles também afirmam que o

projeto do produto final é resultado de ações tomadas nos estados de projeto, os quais são resultados dos estados de transição. Os estados de transição ou o processo de transformação é evidenciado pelas mudanças nos valores dos atributos das restrições de projeto (*relações e regras do domínio de aplicação*), tal como mostrado na Figura 16.

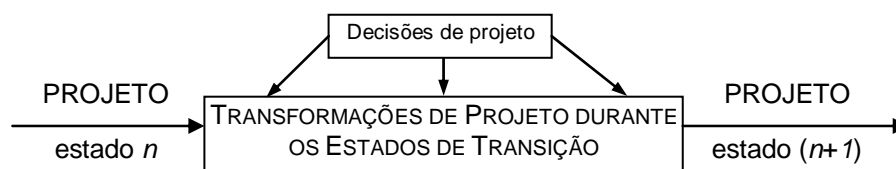


Figura 16. O processo de transformação de dados em informação no projeto.

A diferença entre dados de projeto e informações de projeto está na Tabela 2, a qual resume o que foi discutido no texto acima.

Tabela 2. Comparações entre dado de projeto e informação de projeto.

DADOS	X	INFORMAÇÃO
Está associado à idéia de valor. Por isso, possui um significado de instância de objeto.	X	É idéia do nome associado ao substantivo. Por isso, possui um significado de classe.
Pode ser mutável durante o contexto e dentro do escopo de um processo;	X	É definida e fixa durante o contexto e dentro do escopo de um processo;
É independente do contexto de aplicação;	X	É dependente do contexto de aplicação;
Pode ter muitas representações;	X	Mantém o significado, a semântica;
Possui somente uma interpretação, um ponto de vista;	X	Possui várias interpretações possíveis, dependendo da aplicação;

3.2.2. CONCEITO DE OBJETO E FEATURE

As pesquisas realizadas na década de 90 recomendam que a modelagem de dados deva ser feita com base nos princípios dos modelos orientados a objeto (WARMAN, 1990). Portanto, muitas das companhias fornecedoras de sistemas CAD estão desenvolvendo seus programas baseados no paradigma de orientação a objetos. Elas estão fornecendo recursos e ferramentas que implementam este paradigma. O mais importante de salientar é que, ainda hoje e para o futuro próximo, a escolha por modelos orientados a objeto permanece válida e aceita como a mais indicada para a modelagem computacional de problemas do mundo real.

Seguindo essa tendência, o desenvolvimento de sistemas CAD tem mostrado uma convergência entre os conceitos de sistemas CAD baseados em *features* e métodos de modelagem dessas entidades utilizando os

princípios de orientação a objeto. Várias pesquisas (WARMAN, 1990; KUMAR *et al.*, 1999; DAY, 2000) têm evidenciado esta tendência.

Nas referências utilizadas nessa tese não foi encontrada uma definição universal ou oficial de Orientação a Objeto em CAD. Entretanto, pode-se defini-la assim: O CAD que emprega objetos "inteligentes" (que “sabem” o que são), os quais podem avaliar mensagens e respondê-las (WARMAN, 1990; DAY, 2000).

O termo “CAD Orientado a Objetos” significa que o sistema usa uma representação de objetos do mundo real. Isto é, o usuário não opera num nível semântico de linhas, superfícies ou sólidos primitivos, mas com objetos do mundo real, tais como porta, janela, peça, ressalto, chanfro, rolamento, dobramento, etc.. Além disso, o sistema “*sabe*” como os objetos interagem com os demais. A grande vantagem da utilização de modelagem de objetos em CAD é que os modelos de objetos se auto-validam, porque eles são descrições dos objetos do mundo real. De acordo com o conceito, um objeto “*sabe*” o que ele é.

Um objeto é uma entidade que combina sua estrutura de dados e seu comportamento numa estrutura de dados única. Segundo (MONTENEGRO e PACHECO, 1994), a um objeto qualquer estão associados o seu *estado*, *comportamento* e *identidade*. Esses elementos são definidos conforme abaixo:

- **Estado:** o estado de um objeto é definido pelas propriedades (ATRIBUTOS) que ele possui e pelos correspondentes valores assumidos por estas propriedades.
- **Comportamento:** o comportamento de um objeto é definido pela forma como ele age e reage, em termos de mudança do seu estado e relacionamento com os demais objetos. A ação e reação dos objetos se processam através dos seus MÉTODOS.
- **Identidade:** a identidade de um objeto é definida como a propriedade que o distingue dos demais objetos.

Atualmente, as técnicas de modelagem e programação orientadas a objeto são amplamente empregadas no desenvolvimento de sistemas CAD. Entidades do sistema são divididas em hierarquias de classe. As mensagens instruem aos objetos como eles podem se materializar/instanciar, caracterizando um polimorfismo dinâmico. Novas entidades podem ser

adicionadas, e podem herdar características ou comportamentos encapsulados por classes de objetos. Outra característica da orientação a objeto em sistemas CAD, é que o mecanismo de passagem de mensagens entre objetos reforça os relacionamentos existentes entre modelos sólidos de montagens de conjuntos de peças. Esta modelagem facilita os relacionamentos entre partes do produto.

As técnicas orientadas a objeto fornecem a flexibilidade necessária para o projeto. Os elementos físicos do projeto, seus parâmetros, propriedades, restrições e relacionamentos podem ser representados como objetos, com seus atributos e métodos.

Nos sistemas CAD, seja em termos de Engenharia de Software (*o software em si, do ponto de vista do programador*) quanto da modelagem dos dados (*o modelo de informação, do ponto de vista do usuário*), a modelagem por objetos permite desde a modelagem dos dados assim como a modelagem da própria interface do usuário. Todo o ambiente computacional (*estrutura de dados e interface do usuário*) pode ser construído com base num mesmo modelo de entidades – o Objeto.

Existem algumas implementações de sistemas CAD baseados em objetos, todavia estão mais desenvolvidas no âmbito das interfaces gráficas do usuário, e quando no nível das entidades geométricas, as implementações estão mais desenvolvidas no domínio de suporte ao projeto arquitetônico (ACHTEN e LEEUWEN, 2000).

No que se refere à representação computacional da estrutura de dados dos elementos geométricos, existe um campo de pesquisas ainda cheio de possibilidades de contribuições, que devem se beneficiar dos novos recursos de transmissão, acesso, e compartilhamento de informação via Internet. A necessidade de utilizar esses recursos de redes computacionais integradas é crescente na área de Projeto e Desenvolvimento de Produto, a fim de facilitar a implementação da Engenharia Simultânea.

O contexto computacional de modelagem orientação a objeto usa a definição de objeto para se referir à mesma função que as definições de *features* exercem no contexto computacional de projeto e da tecnologia em sistemas CAD. A idéia de *feature*, usada no projeto de componentes mecânicos e desenvolvimento de produtos, é perfeitamente modelada e

programada por métodos baseados nos princípios de orientação a objeto. Todavia, é importante considerar dois pontos:

- Primeiro que o conceito de objeto é muito mais amplo e subjetivo que o conceito de *feature*, visto que objeto pode ser usado para modelar qualquer coisa do mundo real ou abstrato. Ou seja, pode-se dizer que toda *feature* pode ser modelada como objeto, mas nem todo objeto representa uma *feature*.
- E segundo, que o conceito de objeto não é uma terminologia comum no contexto de Engenharia, visto que sua origem está relacionada à Ciência de Computação.

Pelas considerações levantadas acima, o termo *feature* tem sido usado no contexto computacional de projeto e dos sistemas CAD para representar a idéia de objeto. Assim como o termo objeto é usado em computação, a definição de *feature* é uma abstração que reflete a idéia de objetos dentro do contexto de projeto, como mostrado na Figura 17.

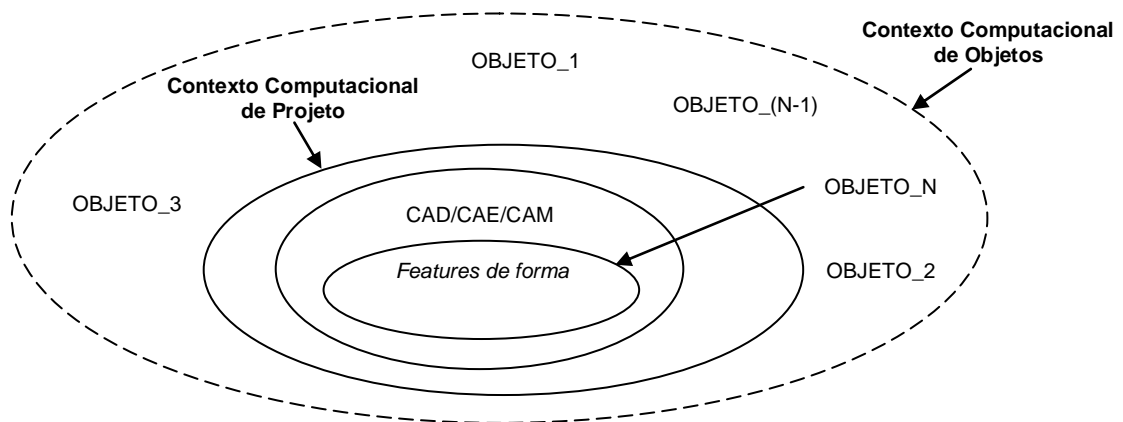


Figura 17. Convergência entre os contextos computacionais de modelagem orientada a objetos e o de projeto baseado em *features*.

Baseado primeiramente no artigo de (CUNNINGHAM e DIXON, 1988) que trata dos primórdios da aplicação do conceito de *feature* na atividade de projetar usando *features* (*Design by features*), fez-se uma revisão cronológica do conceito de *feature*.

A partir da análise e comparação dos conceitos, é possível perceber mudanças do conceito inicial ao conceito atual de *feature*, o qual hoje é tão amplamente difundido na Engenharia.

O conceito de *feature* tem evoluído, e se modificou, estendendo seu significado para além dos processos de inferências e raciocínios associados exclusivamente às atividades de Fabricação. As principais mudanças e avanços ocorrem na modelagem computacional das atividades de Projeto.

Diferentes formas de definir objetos e *features* têm evidenciado as similaridades entre os conceitos de objetos (modelagem computacional de objetos) e *features* (modelagem de entidades do sistema CAD) considerando o contexto de projeto, conforme são listadas abaixo:

Segundo Grayer, em 1976, citado por (MBANG *et. al.*, 2002), a definição original de *feature* foi motivada pelo desenvolvimento de métodos para a geração automatizada de programas de comando numérico, assim: “*a feature descreve uma região que pode ser produzida com uma operação de máquina*”.

Para (CUNNINGHAM e DIXON, 1988), “*feature é qualquer forma ou entidade geométrica que é usada em inferências e raciocínios associados a uma ou mais atividades de projeto e/ou fabricação; por exemplo: ajuste de montagens, função, manufaturabilidade, avaliação/verificação, análise de interferência, projeto de moldes e matrizes e ferramentaria, inspeção, etc.*”.

Para (WARMAN, 1990), “*features ou elementos de projeto são como objetos que possuem um código computacional e podem manipular mensagens*”.

Para (SHAH, 1991), “*features são formas genéricas que os engenheiros associam certas propriedades ou atributos e conhecimento útil para a análise de processos sobre o produto, em outras palavras, as features são vistas como as primitivas de Engenharia*”.

Para (MCGINNIS e ULLMAN, 1992), “*feature é uma característica específica ou particular de um objeto de projeto que contém ou relaciona informação a respeito do objeto. Ela é verbalmente representada na forma de um substantivo*”.

Para (SALOMONS *et al.*, 1993), “*features podem ser como objetos de projeto, pertencendo a uma classe geral, a qual herda propriedades de outras classes*”.

Para (MCMAHON e BROWNE, 1998), “*feature é qualquer elemento ou propriedade geométrico/funcional de um objeto útil para o entendimento da função, do comportamento ou da performance do objeto*”.

Para (BIDARRA e BRONSVOORT, 2000), “*feature é uma representação dos aspectos de forma de um produto, que são mapeáveis para a forma genérica e são significativamente funcionais para alguma fase do ciclo de vida do produto*”.

Para (BRUNETTI e GOLOB, 2000), “*feature é uma unidade de informação (elemento) representando uma região de interesse dentro de um produto. A descrição contém as propriedades relevantes incluindo seus valores e suas relações (estrutura e restrições). Além disso, ela é definida no escopo de uma visão de especificação da descrição do produto com respeito às classes de propriedades e para as fases do ciclo de vida do produto*”.

Segundo citação de (BRONSVOORT e JANSEN, 1993), as definições mais específicas feitas para o termo *feature*, sempre se referem a algum aspecto do modelo do produto ou a aspectos diretamente associadas ou não a uma aplicação específica com ou sem referência à geometria. Abaixo, têm-se as diferentes definições com ênfase nos aspectos citados anteriormente, e que estão destacados em negrito.

Aspectos de geometria e aplicação associados:

- “*a **geometria** que corresponde a uma **operação de usinagem**”;*
- “*uma parte característica ou distintiva/marcante (estereótipo) de uma peça de trabalho, **definindo uma forma geométrica**, a qual ou é específica de um **processo de usinagem** ou pode ser usada para **propósitos de fixação e/ou medição**”.*

Aspectos independentes da aplicação, mas com referência à geometria:

- “*uma **área de interesse** sobre a superfície da peça*”;
- “*uma **região de interesse** no modelo da peça*”.

Aspectos independentes da aplicação, e sem referência à geometria:

- “*um **conjunto de informação** relacionada a descrição da peça*”;
- “*elementos usados na **geração, análise ou avaliação do projeto***”;
- “*um **aspecto funcional** de forma para projeto e fabricação*”.

Segundo (SHAH e MÄNTYLÄ, 1995), aspectos de geometria e semântica podem enfatizados no conceito de *feature*:

Aspectos geométricos:

- “A *feature* é uma **constituente física** de uma peça.”;
- “A *feature* é mapeável para uma **forma genérica**.”

Aspectos semânticos:

- “A *feature* possui **um significado** de Engenharia.”;
- “A *feature* possui **propriedades previsíveis**.”

De forma textual, o conceito de *feature* pode ser sintetizado como:

“*Feature* é toda informação de Engenharia que possui encapsulada a semântica do seu contexto de aplicação, e a qual evolui ou já tem uma conotação/denotação com aspectos de geometria/forma”.

Haja vista que os conceitos de *feature* citados acima se restringiram a pesquisadores estrangeiros, durante o desenvolvimento desse trabalho foi também verificado qual era a compreensão do conceito de *feature* pelos pesquisadores brasileiros no âmbito das áreas de projeto mecânico e fabricação. Este levantamento sobre o conceito de *feature* e as respectivas respostas dos pesquisadores respondentes estão citados no Apêndice 2 dessa tese.

A partir dessas definições, relatos e comentários sobre a modelagem de *feature* e orientação a objeto, pode-se fazer algumas considerações que servirão de base para o desenvolvimento do trabalho de tese.

O conceito de *feature* deve necessariamente contemplar a contribuição de três aspectos: aspectos geométricos e topológicos, aspectos não-geométricos, e aspectos semânticos. A Equação (1) resume isso.

$$\begin{aligned}
 &Feature \Leftrightarrow (Geometria, Topologia) \cup Não-geometria \cup Semântica \\
 &\left\{ \begin{array}{l} (Geometria, Topologia) \cup Não-geometria \Leftrightarrow \text{DADOS} \\ Semântica \Leftrightarrow \text{COMPORTAMENTO} \end{array} \right. \quad (1)
 \end{aligned}$$

A parcela de geometria e topologia do conceito de *feature* representa a parte física da representação do modelo de *feature*. Essa parcela é independente, exata e quantitativa. Associada à parcela de geometria e

topologia tem-se a parcela de não-geometria, as quais juntas constituem a parte de dados da *feature*, os quais são implementados através de atributos.

A parcela semântica, por sua vez, representa a parte abstrata da representação do modelo de *feature*. Essa parcela é dependente, subjetiva e qualitativa. Do conceito de *feature*, sabe-se que semântica é dependente do nível de abstração, do tipo de produto, e do tipo de aplicação (SHAH e MÄNTYLÄ, 1995). Sabe-se ainda que a semântica é responsável pela captura das restrições e dos comportamentos de uma *feature*, os quais são implementados através de métodos.

Quando a parcela semântica infere apenas sobre a parcela de geometria e topologia do modelo de *feature*, pode-se concluir a caracterização de uma *feature* de forma, de acordo como definido em (SHAH e MÄNTYLÄ, 1995). Pode-se afirmar que a *feature* de forma seja o tipo de *feature* de mais baixo nível, e por isso justifique porque tenha sido o primeiro e mais aplicado.

Quando a parcela semântica infere sobre o contexto de aplicação de não-geometria, diferente de somente geometria e topologia, tem-se a caracterização de uma *feature* de aplicação, por exemplo: *feature* de montagem, *feature* funcional, *feature* de análise, etc..

Mostrando o conceito de objeto, tem-se pela Equação (2) o seguinte:

$$\begin{aligned} \text{Objeto} &\Leftrightarrow \text{Atributos} \cup \text{Métodos} \\ \left\{ \begin{array}{l} \text{Atributos} \Leftrightarrow \text{DADOS} \\ \text{Métodos} \Leftrightarrow \text{COMPORTAMENTO} \end{array} \right. & \quad (2) \end{aligned}$$

Fazendo uma comparação das Equações (1) e (2), pode-se concluir que:

$$\begin{aligned} \left\{ \begin{array}{l} (\text{Geometria, Topologia}) \cup \text{Não-geometria} \Leftrightarrow \text{Atributos} \Leftrightarrow \text{DADOS} \\ \text{Semântica} \Leftrightarrow \text{Métodos} \Leftrightarrow \text{COMPORTAMENTO} \end{array} \right. & \\ \Downarrow & \\ \text{Objeto} \equiv \text{Feature} & \quad (3) \end{aligned}$$

Comentando a Equação (3), o sinal de identidade pode ser interpretado como uma herança (*para usar a terminologia de orientação a objetos*) existente entre a representação de objetos e de *features*. Em suma, as *features* são objetos e podem ser modeladas perfeitamente como tal.

3.3. CONSIDERAÇÕES SOBRE AS PROPOSIÇÕES DA TESE

Nesse ponto da tese, o texto das secções precedentes conduz à necessidade de delinear os passos seguintes, e mostrar as ligações e relacionamentos entre as questões já comentadas e descritas.

Respondendo algumas das perguntas feitas na secção 1.2:

- O modelo de objeto é o modelo mais adequado e atual para ser utilizado na modelagem das informações e dados de projeto, pelas razões colocadas em toda a secção 3.2.2 e concluídas na Equação (3). No caso do domínio de Engenharia, viu-se que o modelo de *features* é também o que tem de mais atual e que correspondente ao modelo de objetos. Isso responde à questão 5.
- Nessa tese, as *features* de interesse são enquadradas como um tipo de *features* de projeto, a saber, as ***features de fase do processo de projeto***. A parcela semântica leva em conta a captura das informações referentes a cada uma das fases do PPP. Esse enquadramento implica considerar o caso de *features* que dependendo da fase de projeto, as mesmas não possuem referência/associação a nenhuma geometria e/ou topologia, isto é, possuem somente referências a características não geométricas. Esse caráter de não exigir uma associação a características geométricas amplia o conceito de *feature*, e traz uma nova visualização para a Figura 17, mostrada aqui na Figura 18. Isso responde à questão 3.

O Capítulo 4 que segue, vai introduzir a metodologia de *featurização*, a forma de como implementar a *featurização* no processo de projeto de produto. Essa discussão da *featurização* refere-se à questão 4, e assim mostrar através de um estudo de caso, como é possível estender as funcionalidades dos sistemas computacionais de suporte ao projeto, mais especificamente em Sistemas CAD.

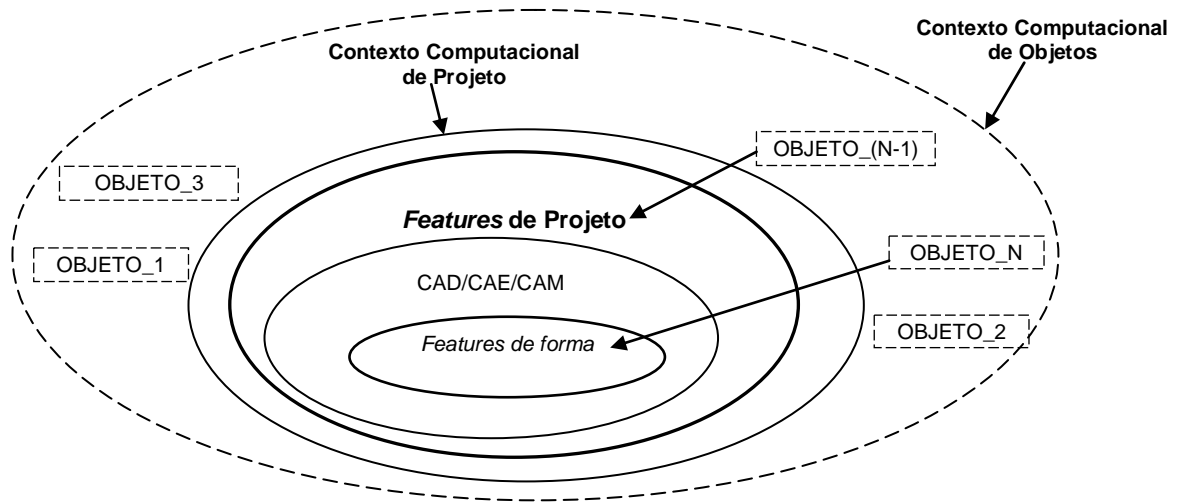


Figura 18. Enquadramento da visão de *feature* de projeto na tese.

Capítulo 4

Sistemática de *Featurização* no Processo de Projeto de Produto

4.1. INTRODUÇÃO

Acompanhando o desenvolvimento do processo de projeto, percebe-se uma evolução da informação e dados, caracterizada por uma transição de aspectos mais abstratos, imaginários e ideais em direção a aspectos cada vez mais concretos, reais e factíveis.

No projeto informacional definem-se as especificações de projeto e no projeto conceitual elas são processadas, resultando numa concepção para o problema de projeto. Traduzido na visão do processo de projeto, isso é um salto imenso; o qual embora seja identificado pelos pesquisadores, mas que pela própria natureza como se processa, ele é caracterizado como nebuloso, criativo e cognitivo. Em suma, é um processo não determinista.

A idéia e o anseio de empregar e enxergar uma sistemática para esse processo, vai da necessidade de automatizá-lo e enquadrá-lo num modelo de implementação computacional factível, não ideal. A razão disso é identificar

os processos cognitivos presentes, os quais tornariam visíveis os passos intermediários existentes nas interfaces entre as fases do processo de projeto como um todo.

Em sistemas CAD, o conceito empregado e indicado para tanto seria o modelo baseado no conceito de *features*, conforme discutido no Capítulo 3.

Seguindo essa tendência de aplicações baseadas em *features*, essa tese traz a proposta de *featurização* de produtos mecânicos como uma sistemática para documentação dos dados e informações associadas ao produto.

Um conjunto de conceitos importantes para o entendimento dessa sistemática e que estão no contexto da tese são definidos e discutidos no tópico 4.2. Nesse item também se comenta sobre a organização dos dados e informações empregados por essa tese. No tópico 4.3 são comentados os aspectos relacionados à preparação e implementação da *featurização*, enfatizando quais pontos são importantes de serem considerados. No tópico 4.4, descreve-se o estudo de caso realizado para a verificação da aplicabilidade da sistemática. E finalmente, as conclusões e considerações finais sobre a sistemática são citadas e comentadas.

4.2. CONCEITOS E DEFINIÇÕES

O conceito original de *featurização* advém dos trabalhos de (SHAH e MÄNTYLÄ, 1995). Segundo eles, o termo *featurização* refere-se ao processo de identificação e formalização das *features* de uma peça para uso em alguma aplicação específica.

A metodologia esboçada por Shah e Mäntylä preceitua, inicialmente, a análise isolada do produto e do processo. E na seqüência, numa fase posterior finalizada a análise, conceber a *featurização* do produto conciliando as duas visões, e assim explicitando no modelo de CAD baseado em *features* – o qual seria o modelo do produto nesta fase – os relacionamentos entre as informações de projeto.

A partir do esboço de Shah e Mäntylä, essa sistemática de *featurização* foi expandida e detalhada nessa tese. Ela também foi implementada através de um estudo de caso apresentado no tópico 4.4, como uma primeira

verificação da adaptação e implementação, e com vistas a avaliar as vantagens e melhorias em se estender tal sistemática genérica no processo de modelagem de peças/produto usando diferentes sistemas CAD comerciais.

Vale ressaltar que o modelo do produto criado pela aplicação dessa sistemática pode ser descrito como uma combinação de modelo paramétrico com um modelo de *feature* definida pelo sistema CAD usado na implementação da sistemática.

Tecnicamente, essa questão é conhecida como abrangência do modelo baseado em *features*. Ou seja, algumas informações no modelo do produto, a exemplo daquelas capturadas no nível dos perfis bidimensionais, ainda estão sendo modeladas num nível básico das entidades do sistema CAD – que corresponde à parte do modelo paramétrico. Enquanto que outras partes já estão definidas como operações e procedimentos automatizados em definições de *features* nativas do próprio sistema ou mesmo através de recursos de definição de *features* compostas (*ou seja, definidas conforme a visão do usuário*) – e essa parte corresponde ao modelo baseado em *features*. A proposta da *featurização* é usar os recursos disponibilizados pelo sistema CAD específico, para tornar o modelo do produto o máximo possível baseado em *features*. Isso é conseguido através da personalização dos recursos do sistema CAD, conforme é mostrado no estudo de caso desenvolvido.

Nessa *featurização* das informações de projeto, a organização dos dados e informações do produto segue a mesma adotada pelos modeladores sólidos dos principais sistemas CAD. Na estrutura desses sistemas, é possível abstrair e modelar o produto como sendo constituído pela seguinte estrutura hierárquica: perfis bidimensionais parametrizados geram *features* geométricas definidas no sistema CAD, que por sua vez geram as peças do produto, e finalmente as mesmas são colocadas em arquivos de montagem. A Figura 19 resume essa estrutura hierárquica.

Essa é uma organização de dados consagrada, aceita e adotada pelos principais sistemas CAD comerciais utilizados pela indústria, e espelha bem a realidade da organização dos dados e informações no contexto do projeto mecânico, e serve muito bem aos requisitos da sistemática de *featurização* do produto.

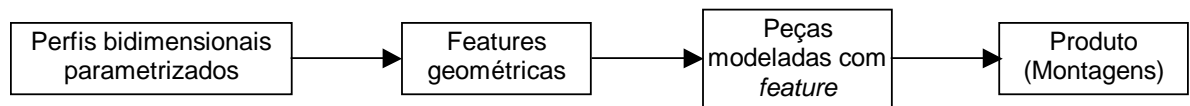


Figura 19. Estrutura hierárquica da informação do produto no sistema CAD.

E por enxergar essa organização de dados já programada dentro dos sistemas CAD, essa tese a sugere como sendo a mais adequada nesse processo de integração entre a sistemática de *featurização* e os diferentes sistemas CAD empregados para sua implementação.

4.3. SISTEMÁTICA DE *FEATURIZAÇÃO*

Os critérios adotados para a *featurização* do produto podem se basear em diferentes pontos de vistas possíveis de visualizar o produto e que são de interesse para a definição do mesmo.

A *featurização* se aplica na documentação do produto. Essa documentação é focada no detalhamento das informações e dados das peças que compõem o produto, principalmente na sua fase final do projeto detalhado. A partir dessa primeira aplicação centrada na fase final do processo de projeto, como usualmente é feita, é intenção também utilizar e estender essa sistemática para todas as fases do processo de projeto.

A Figura 20 ilustra uma visão completa dos passos de implementação da *featurização* do produto e uma melhor compreensão da mesma. Esses passos são comentados e discutidos no texto que segue.

É condição necessária para dar início ao processo de *featurização* do produto, conhecer o máximo possível do produto. Para tanto, a fase inicial consiste em colher informação sobre o produto. Essa informação pode se apresentar e estar disponível em diferentes formatos:

- **Verbal:** através de entrevistas ou conversas informais com todas as pessoas envolvidas com o ciclo de vida do produto, as quais correspondem ao que pode ser chamado de clientes. Todos eles possuem informações importantes em alguma fase do ciclo de vida do produto, desde a concepção até o descarte do produto, passando por fabricação, operação e manutenção;

- **Impresso:** documentação textual gerada à mão, datilografada, ou mesmo textual gerada em computador, tais como relatórios técnicos, questionários estruturados, listas de verificação (*checklists*), etc.;
- **Digital:** arquivos de formato computacional nativo referente a algum software específico usado pela empresa. Tais informações podem ser os próprios desenhos técnicos produzidos para documentação do projeto do produto, e toda a informação que normalmente é inerente a este tipo de documentação, tais como: simbologias de tolerância, acabamento superficial e processos de fabricação, notas gerais de desenho, etc.

Das informações inicialmente colhidas sobre o produto, aquelas relacionadas à geometria do produto são as mais importantes. Pode-se afirmar que, num primeiro instante, os desenhos técnicos do produto são os recursos mais importantes para dar início ao processo de *featurização*.

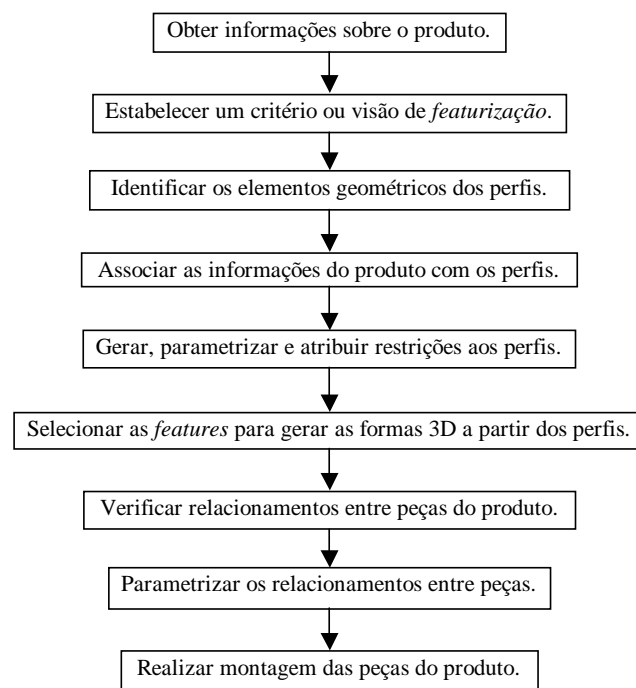


Figura 20. Os passos da *featurização* do produto.

Nessa tese, o critério utilizado na parametrização e *featurização* das peças, baseia-se primeiramente nas intenções do projetista em obter a forma final desejada da peça. Conjuntamente a esse critério, busca-se na seqüência uma associação e registro explícito também dos aspectos

funcionais e/ou das intenções e da sua modelagem sólida no sistema CAD. E assim, pretende-se armazenar no modelo do produto gerado pelo sistema CAD, as informações não somente geométricas, mas também intenções, funções e outros dados mais abstratos que fazem parte da informação necessária a uma completa descrição associada ao produto.

A partir dos desenhos, prossegue-se com a identificação da informação candidata a ser capturada no arquivo do modelo CAD. A fase de identificação contempla a seguinte classificação de entidades: perfis de construção/geração das formas das *features*, as *features*, e as peças que compõem o produto. Visto que também os sistemas CAD contemplam a modelagem da montagem das peças que constituem o produto, também é possível estender a *featurização* a este nível, completando assim o conjunto de informações e dados citados na Figura 19.

Essa fase consiste em realizar análise dos desenhos do produto, interpretando as informações contidas nele com o seguinte enfoque:

- Abordar inicialmente as peças com a identificação de aspectos relacionados à modelagem sólida da geometria, topologia, e dimensões no sistema CAD, o que é entendido como explicitar aspectos e intenções de projeto relacionadas com a geração e decisão da forma final do produto;
- Identificar regiões de interesse, seja no nível de peça ou do produto, como um todo. Se possível, é importante discernir bem o nível de captura da informação, se é para o produto ou somente para a peça;
- Identificar possíveis detalhes que sejam característicos no produto, tais como elementos geométricos repetitivos ou recorrentes na forma do produto, ou elementos que possuem uma característica funcional marcante e decisiva.

Observando o produto com ênfase nesses itens citados acima, objetiva-se identificar relacionamentos entre o critério de *featurização* e a forma das peças que o compõe, algo como um estereótipo (*caricatura*) do produto quando observado segundo um ponto de vista específico.

Como diretriz básica, deve-se atentar para os requisitos assinalados por (SHAH e MÄNTYLÄ, 1995), na identificação das informações candidatas ao

critério especificado para a *featurização* do produto, a saber:

- Os elementos identificados servem como uma função do produto;
- Eles atuam como formas ou aspectos recorrentes em várias versões do produto. A identificação desses aspectos sinaliza para a decisão de formalizar ou não os elementos como *features*;
- Eles são passíveis de parametrização, ou descrição por um conjunto de relações geométricas e/ou um procedimento, ou mesmo representação baseadas em *features*;
- Eles pertencem ao vocabulário comum do projetista ou do contexto em questão. Esse requisito favorece a captura do significado e/ou intenção de quem instancia a informação de projeto e melhora o entendimento de quem usa e/ou processa a informação dentro do contexto ao qual ela se aplica.

A identificação e explicitação dos elementos significativos para o produto são realizadas no sentido de satisfazer essas condições. É lógico, que a natureza genérica desses requisitos torna possível uma interpretação flexível em atender a todos, ou a maioria deles, ou mesmo adicionar novos requisitos dependendo do tipo de produto e informação específica em estudo. Deve-se considerar esses requisitos como diretrizes ou recomendações básicas para a caracterização da informação capturada, sem que nada impeça que os mesmos sejam ou não satisfeitos na sua maioria.

Com base na forma do produto capturada e armazenada no modelo CAD, parte-se agora para a associação das informações mais relacionadas a diferentes pontos de vistas do produto. A associação dessas informações com as formas identificadas na fase anterior, além de enriquecer o modelo do produto, gera uma ligação e explicita os relacionamentos entre as evoluções sofridas pela informação nas diferentes fases do processo de projeto.

A idéia é demonstrar que é possível usar a parametrização para elevar o nível de abstração na manipulação das informações do produto colocando o mais próximo possível do que seria um modelo puramente baseado em *features*. A diferença é que muita da semântica armazenada no modelo CAD não está automatizada, mas foi criada manualmente utilizando recursos já disponíveis nos sistemas CAD utilizados.

4.4. ESTUDO DE CASO

Os procedimentos que levaram à implementação dessa sistemática seguiram as recomendações e diretrizes básicas descritas no item 4.3. Nesse estudo caso, a natureza do projeto consiste de uma betoneira, máquina tradicionalmente usada na indústria da construção civil. O projeto dessa máquina caracteriza-se como um projeto de rotina, onde os requisitos e princípios de solução utilizados já são, de certa maneira, amplamente conhecidos. Vale salientar que a *featurização* nesse estudo se resumiu a documentação do projeto final da betoneira, com ênfase na facilidade de gerar as formas das peças do produto. Por isso, a avaliação da sistemática deve se concentrar nesse aspecto.

A coleta do material desse estudo de *featurização* iniciou pelos desenhos bidimensionais da betoneira mostrada na Figura 21(a). Esse produto faz parte da linha de produção de Metalúrgica Erwino Menegotti Ltda., situada em Jaraguá do Sul – SC. Embora a sistemática implementada neste estudo de caso contemple apenas o caso de um projeto já definido, entende-se que é factível estender essa sistemática e inseri-la como uma sistemática geral de documentação das informações e dados das fases do processo de projeto, como a idéia colocada em (CUNHA e DIAS, 2000b) e (SILVA J., 2001).

Como exemplo de aplicação da sistemática de *featurização*, considere a peça da coluna maior da betoneira na Figura 21(b), para mostrar em detalhes as decisões feitas para a *featurização*, levando em conta todo o conjunto de recursos disponibilizados pelo sistema CAD usado.

A Figura 22 mostra que a peça é constituída por um perfil vazado de seção transversal retangular com uma determinada espessura e alguns detalhes de cortes realizados ao longo da peça. Fazendo-se a análise sobre as possíveis formas características dessa peça, foram eleitos os perfis da seção transversal retangular e do corte circular na extremidade da peça. Parte-se então para a parametrização, evidenciando os parâmetros importantes na especificação de perfis desse tipo.

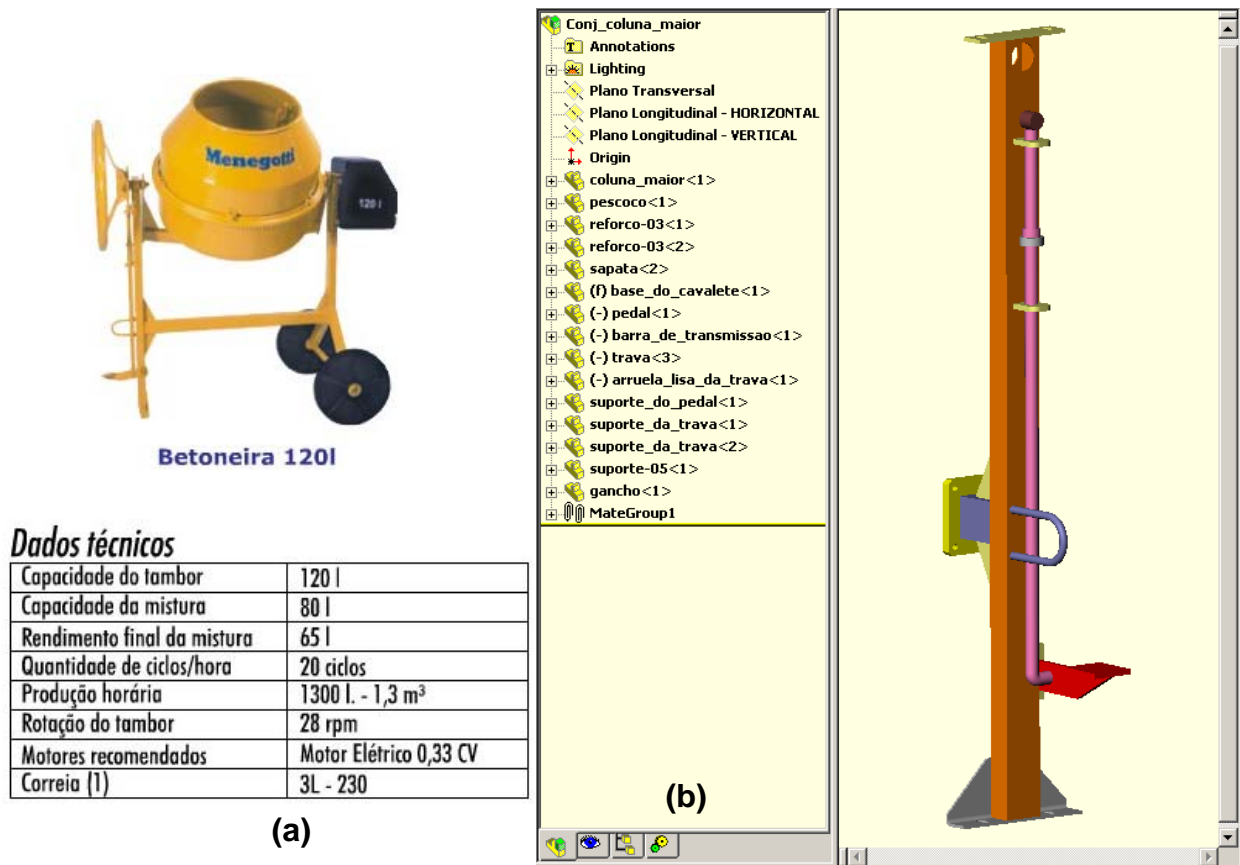


Figura 21. O produto e sua especificação técnica e o conjunto estudado.

Importante ressaltar que estes perfis podem ser agora armazenados numa biblioteca de perfis. Esse recurso, normalmente, já vem disponibilizado dentro do próprio sistema CAD.

Outra alternativa quando não se tem disponível este tipo de serviço, é usar um banco de dados e associar os parâmetros dos perfis parametrizados com variáveis de controle desses valores dentro do sistema CAD, conforme foi avaliado e testado no trabalho de (SILVA, 2001). Isso geralmente é conseguido através de programação numa API fornecida pelo sistema CAD. A desvantagem desse segundo método comparado ao primeiro, é que ele exige conhecimentos de programação e da API do CAD.

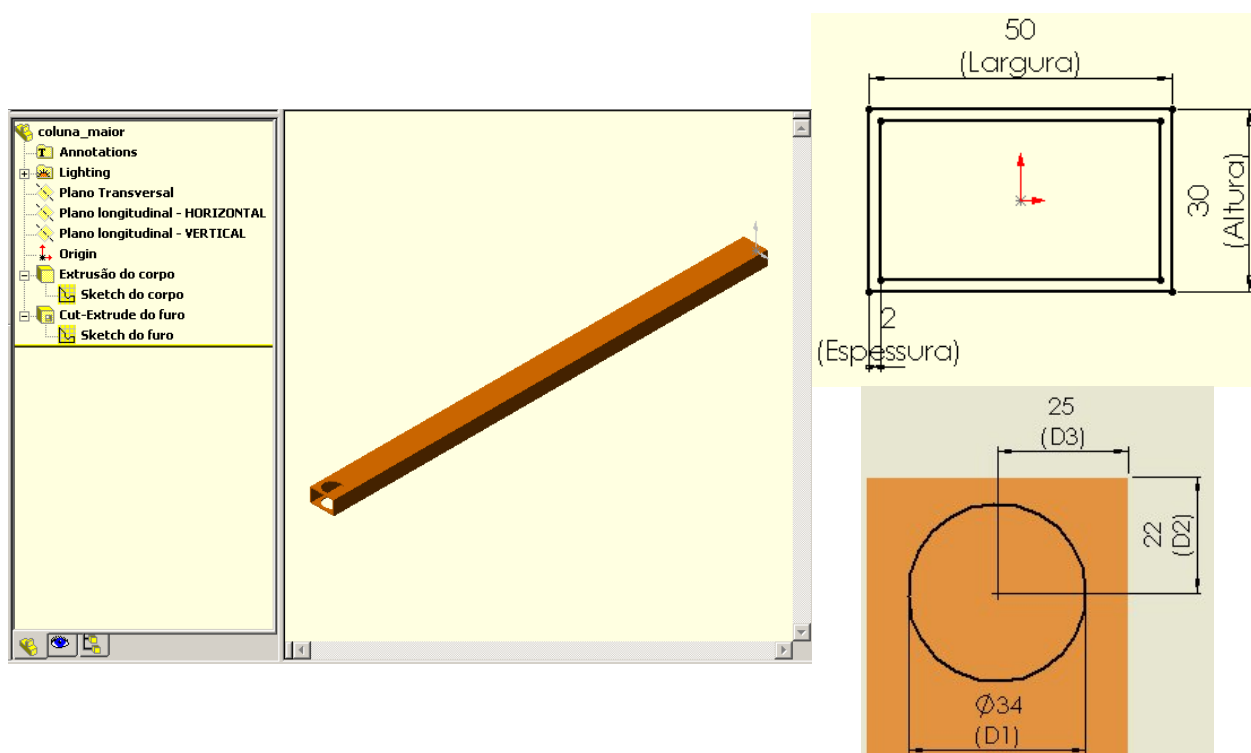


Figura 22. Peça e perfis parametrizados na geração de formas usando *features*.

Neste estudo de caso foi implementada a primeira alternativa, utilizando para demonstração desse recurso o sistema CAD SolidWorks da *SolidWorks Co.*. A Figura 23 mostra uma tela do programa evidenciando a biblioteca de perfis, e também a organização da informação em termos de peças e de *features* armazenadas da betoneira.

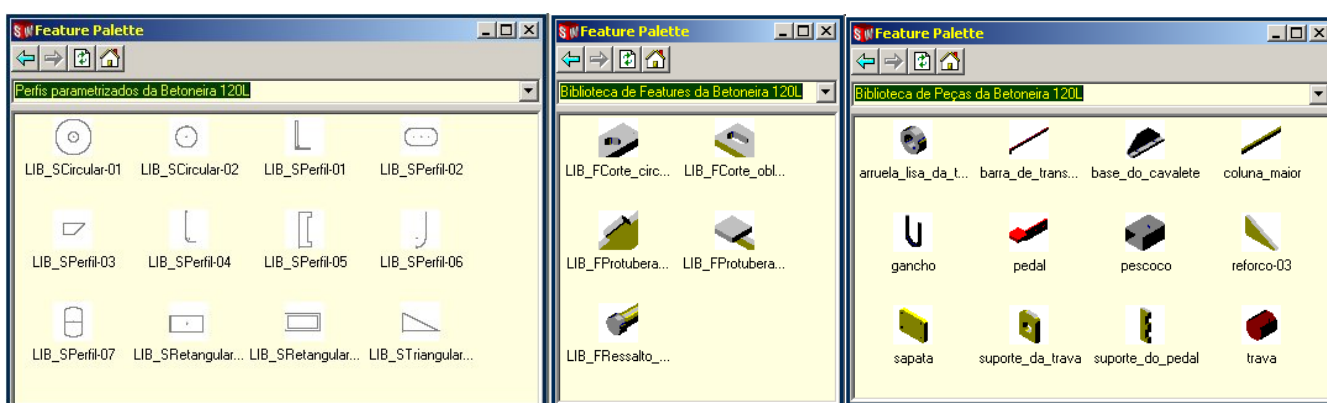
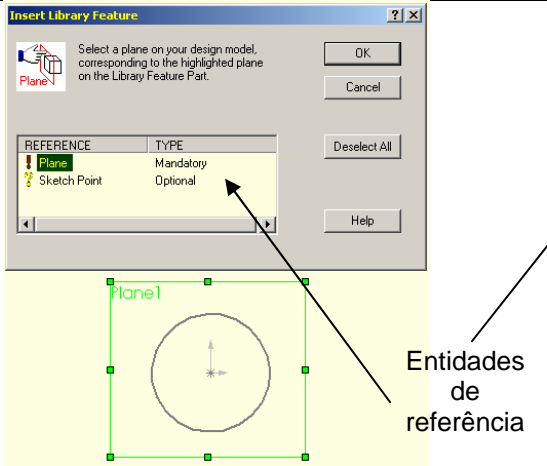
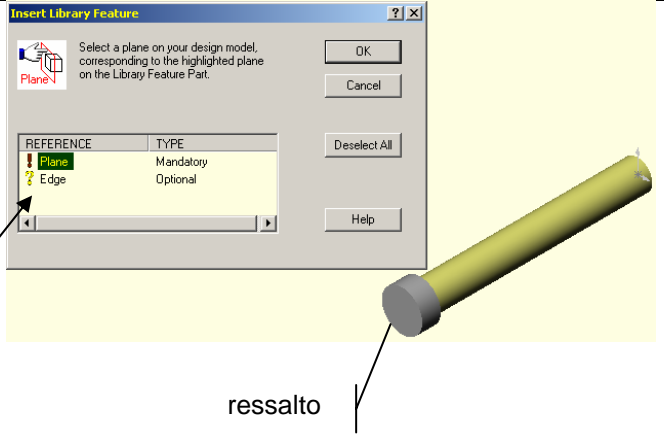


Figura 23. Biblioteca de perfis do conjunto coluna maior da betoneira 120L.

O benefício de utilizar a sistemática de *featurização* no estudo realizado, pode ser exemplificado no processo de geração das peças do produto. Na

Tabela 3, considera-se a barra de transmissão como exemplo para realçar alguns aspectos do processo de *featurização* de um produto, usando os recursos ainda não totalmente automatizados ou encapsulados por uma operação puramente baseada em *features*.

Tabela 3. Exemplo da geração de formas das peças com restrições e significados semânticos associados e encapsulados pela sistemática de *featurização*.

<p>Utilizando o perfil circular da biblioteca de perfis, insere-o no ambiente de modelagem. Abaixo, mostra-se o sistema CAD habilitando os recursos de posicionamento e orientação do perfil e possibilitando escolher as entidades de referência. Quando o perfil estiver instanciado, pode-se nomeá-lo com um termo significativo para o contexto da peça ou da operação de modelagem, conforme é visto na Figura 22.</p>	<p>Modela-se a <i>feature</i> ressalto, definida pelo usuário do sistema e armazenada na biblioteca de <i>features</i> com todas as restrições e significado de engenharia pertinentes ao vocabulário associado à idéia ressalto. Da mesma forma, o sistema CAD habilita os recursos de posicionamento pertinentes a um ressalto, ou seja, qual o plano (<i>o mesmo pode ser uma face da feature base</i>) e qual a aresta de anexação.</p>
 <p>Entidades de referência</p>	 <p>ressalto</p>

Essas operações de modelagem demonstram o poder de encapsular no modelo geométrico, a semântica associada à forma geométrica. Pois assim,

tem-se a forma geométrica gerada com as características exigidas pelo contexto de aplicação de forma automática, sem a necessidade de introduzi-las manualmente. O controle e a manutenção da semântica também ficam melhorados, seguindo o que é recomendado e defendido pelos trabalhos de (BIDARRA, 1999) e (CASE e HOUNSELL, 2000).

No caso do ressalto da Tabela 3, o mesmo sempre ficará com o valor correto independente se o diâmetro da *feature* base é modificado. Ou se o comprimento da *feature* base também é modificado, movendo a face de referência do ressalto, assim mesmo o ressalto deve continuar anexado à sua face de referência. Esse exemplo mostra como são e podem ser capturadas as informações geométricas e também não-geométricas numa *feature*.

A idéia é também evidenciar a possibilidade de implementação dessa sistemática em quaisquer sistemas CAD que satisfaçam alguns requisitos e características mínimas.

Para tanto, alguns requisitos são exigidos desses sistemas para se verificar uma produtividade e conveniência de implementação da sistemática dentro de um escritório de projeto de médio/grande porte. A saber:

- Permitir a modelagem tridimensional do produto;
- Possibilitar a parametrização do produto baseado em features;
- Oferecer recursos de edição, criação e manipulação de restrições geométricas e dimensionais, além de variáveis e equações associadas às entidades do sistema CAD. É também interessante que o sistema disponibilize recursos de posicionamento, orientação e referenciação das entidades geométricas, como exemplificados na Tabela 3;
- Oferecer recursos de personalização dos termos dados às entidades explicitadas no modelo geométrico do sistema CAD, sendo assim possível nomear as *features* com nomes mais próximos à semântica de aplicação.

O atendimento desses requisitos é resolvido de forma particular e peculiar por cada um dos sistemas CAD comerciais disponíveis no mercado. A aplicabilidade desses recursos só fica explícita, e passível de uso à medida que vai se explorando, interagindo, dominando e entendendo o sistema como um todo.

4.5. CONSIDERAÇÕES PARA A TESE

A sistemática de *featurização* explicada nesse capítulo evidencia, com base nos exemplos, a importância do conceito de *features* corretamente implementado num modelo de *features* aplicado em sistema CAD.

E para a tese, a sistemática de *featurização* evidencia a possibilidade de estender os seus passos, conforme descritos na secção 4.3, para dentro de todas as fases do processo de projeto, e assim integrando um pouco mais o sistema CAD ou outros sistemas de apoio ao processo de projeto, com as sistemáticas de projeto.

O Capítulo 5 chama a atenção desse fato e propõe a implementação do processo de *featurização* a partir da modelagem de informações de projeto nas fases do processo de projeto.

Capítulo 5

Features nas Fases do Processo de Projeto de Produto

5.1. INTRODUÇÃO

A idéia principal desse trabalho é determinar o conjunto de informações importantes para cada fase de projeto, as quais são modeladas e representadas como *features*, e assim obter uma identidade da fase. Esse conjunto de informações características seria chamado, ou interpretado como uma **identidade da fase de projeto** correspondente.

Para a definição das *features* que representariam a identidade das fases de projeto, tem-se a necessidade de sistematizar o tipo de informação que ocorrem nas fases; e a forma como essas informações são fornecidas, representadas e armazenadas no computador.

As definições de um protocolo e de uma terminologia padronizada e apropriada de conceitos podem auxiliar na sistematização da modelagem de informação. Essa sistematização deve priorizar um levantamento de quais informações são importantes, quais as formas de representação, e o

estabelecimento da estrutura de dados adequada para cada fase de projeto, a qual resulte em informações geométricas e não-geométricas de projeto.

Para a representação desse conjunto de informações, propõe-se a aplicação de uma abordagem baseada em modelos de *features* de projeto para toda a estrutura de informação do produto e do processo, que nessa tese acontece através dos passos descritos na sistemática de *featurização* do Capítulo 4.

5.2. TRANSFORMAÇÃO DOS DADOS DE PROJETO

Segundo (BACK, 1983), o projeto é considerado como um sistema que transforma “informações”. A razão de capturar os dados de projeto de um produto é poder mostrar o desenvolvimento e a propagação das restrições de projeto, a relação entre as *features* de projeto e as restrições, e identificar as necessidades de representação computacional no projeto mecânico. Isso significa obter um retrato da história do projeto como informação para o produto.

No projeto de um componente, algumas restrições sobre sua forma e função podem ser dadas no início do problema. Outras restrições aparecem a partir do domínio de conhecimento do projetista. Outras são derivadas a partir de cada decisão de projeto tomada. É o relacionamento desses três tipos de restrições de projeto que formam o refinamento evolucionário de um componente. Esta evolução das informações de projeto quando é traçada em detalhes, fica claramente demonstrada com ênfase nessas restrições, e a fonte de cada restrição de projeto.

Outra razão de querer entender as transformações dos dados de um projeto é esclarecer os conceitos e os relacionamentos de **restrição de projeto** e **features de projeto** entre esses termos. Esses conceitos são, às vezes, usados de forma inconsistente. Para traçar a história do projeto de um componente específico, suas restrições e *features* têm de ser bem identificadas e esclarecidas, e as relações entre elas desenvolvidas.

Para implementação da proposta colocada nesse trabalho, o sistema CAD é a ferramenta principal. A ênfase neste trabalho será sobre os sistemas CAD, ou seja, pretende-se pensar em soluções que possam ser incorporadas

aos recursos computacionais normalmente existentes num sistema CAD qualquer, de forma a melhor integrá-lo à atividade de projeto.

No início da utilização dos sistemas CAD nas atividades de projeto, a maior preocupação se concentrava na automação dos processos de geração de desenho. O objetivo era aumentar a produtividade e a documentação do projeto detalhado do produto. Ou seja, a fase de documentação ficava no final, e isolada das demais fases do processo de projeto, como mostra a Figura 24.

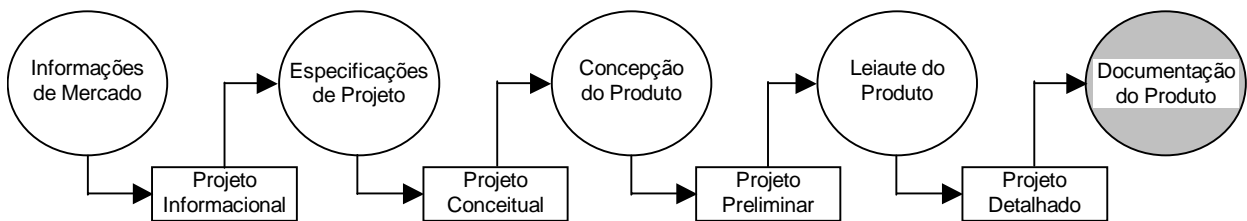


Figura 24. Fases de projeto com as respectivas entradas e saídas (Baseado em OGLIARI, 1999).

Conforme mostrado pela Figura 25, é preciso que o sistema CAD armazene as informações de projeto desde o seu início. Nesse estudo sugere-se que a fase de documentação do projeto ocorra desde a fase Informacional/Conceitual até o projeto detalhado. Dessa forma, tem-se a documentação do produto fazendo parte de todas as fases do projeto, e não somente do final.

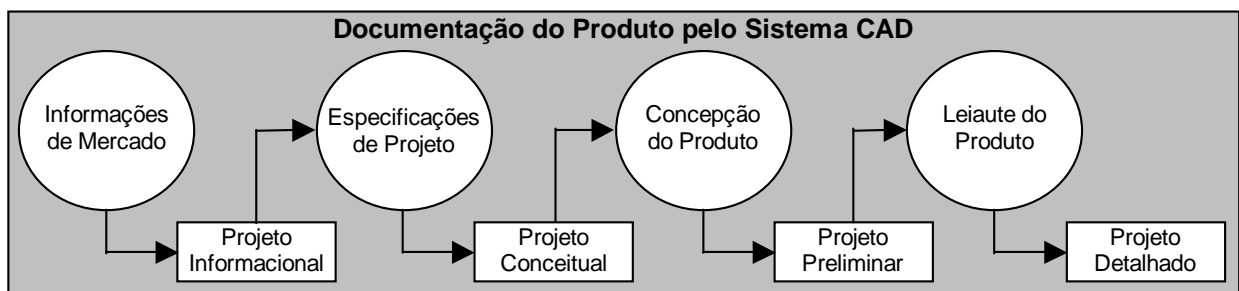


Figura 25. Fases de projeto e documentação do produto.

A partir dessa visualização diferenciada da ordem das fases de projeto, uma sistemática de *featurização*, conforme descrita no Capítulo 4 e integrada ao sistema CAD, é proposta na seção 5.3 e implementada no Capítulo 6 de resultados da tese.

5.3. FLUXO DE INFORMAÇÃO NO PROJETO

O reuso, a troca e o compartilhamento de dados e informações de Engenharia no processo de projeto, necessitam de bases computacional e metodológica consistentes para serem satisfatórias e eficientes.

A fim de facilitar e otimizar a integração de qualquer ferramenta computacional na atividade de projeto sugere-se a aplicação de uma sistemática de projeto, com fases bem definidas.

Assim sendo, com o objetivo de introduzir o uso computacional para suportar as informações geradas nas atividades de projeto, as fases de projeto consideradas nesse estudo são as que estão representadas por retângulos, como é mostrado na Figura 25.

(SHAH e MÄNTYLÄ, 1995) sugerem que uma linguagem para o desenvolvimento de produto deve considerar dois elementos: o *processo* e o *produto*. Confrontando a Figura 25, com o que dizem Shah e Mäntylä, pode-se visualizar que os processos são as *fases de projeto*, e dão uma indicação sobre quais processos os dados do produto estão sujeitos na respectiva fase. Já as *entradas* e *saídas*, representadas por círculos, constituem os estados do projeto, e dão uma indicação dos dados e informações do produto, ou seja, do estado do produto. Isso é mostrado na Figura 26.

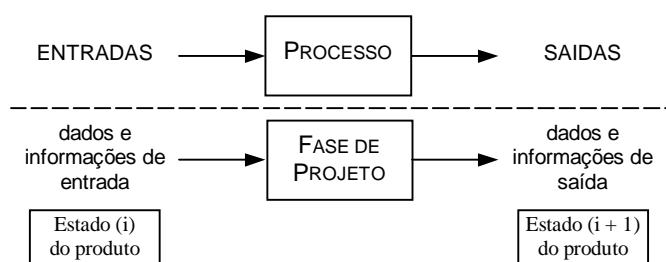


Figura 26. Processos x Fases de Projeto – Produto x Dados e Informações.

Nesta linguagem de desenvolvimento do produto, os processos informam sobre a evolução do produto através dos estados de projeto. Cada ação de projeto o modifica. Ou seja, os estados de projeto são como fotografias instantâneas tiradas durante seu desenvolvimento. Então, é importante caracterizar conceitos que estão associados a cada fase do processo de projeto, e as suas correspondentes transições.

Assim, as diferentes fases e transições são classificadas como:

- No *Projeto Informacional*, as **informações de mercado**, caracterizadas pelos **interesses** ou **manifestações** das pessoas que direta ou indiretamente relacionam-se com o produto, transformando-se em **especificações de projeto** que caracterizam os **problemas técnicos** e as **restrições** a serem resolvidas.
- No *Projeto Conceitual*, os valores quantificados das **especificações de projeto**, transformando-se numa solução de concepção simplificada para o problema de projeto. Essa solução é caracterizada pelas principais **funcionalidades** e **princípios de solução**, e representados através de **esquemas** ou **esboços** das formas aproximadas do projeto.
- No *Projeto Preliminar*, as **funções do produto** e **princípios de solução**, transformando-se num **leiaute do produto**, caracterizado e representado pelos **elementos** que constituem o produto e suas **principais geometrias** e **formas**.
- No *Projeto Detalhado*, tem-se o detalhamento do **leiaute** do produto. Nesta fase, as **dimensões**, **tolerâncias**, **materiais** e a **forma** dos componentes individuais de projeto são especificados em detalhes para as subseqüentes fases de fabricação e produção do produto.

As entradas e saídas de cada fase de projeto, assinaladas em negrito nos comentários anteriores, são compostas por informações geométricas e não-geométricas. Estas informações podem ser definidas como *features* de projeto, e podem ser organizadas de acordo com os princípios de orientação a objeto, a fim de suportar a reusabilidade da modelagem das informações de uma estrutura computacional.

A proposta é de que a evolução do projeto seja baseada: na sistemática de projeto da Figura 25, e a modelagem das informações de projeto importantes das fases de projeto se baseie na modelagem de *features* de projeto e na modelagem de objetos.

Assim, em cada fase, as *features* de projeto terão dados relacionados à sua evolução dentro do projeto. Essas *features* devem evoluir a partir dos esboços de formas do projeto e grandezas numéricas extraídas numa fase

informacional e conceitual, para *features* de detalhamento durante o projeto detalhado. A modelagem de objetos, por sua vez, fornece os meios e princípios para definição de estruturas de dados reusáveis e adequadas à captura do comportamento, da função e da intenção do projetista.

Resultante de um primeiro esforço de implementação, o Capítulo 6 identifica e classifica os objetos de projeto e as *features* de fase do PPP.

A Figura 27 faz uma extensão da Figura 25, adicionando quais informações estariam no contexto da fase, as formas de representação mais comumente assumida em cada tipo de informação, e uma idéia da evolução dessas informações entre as fases do PPP.

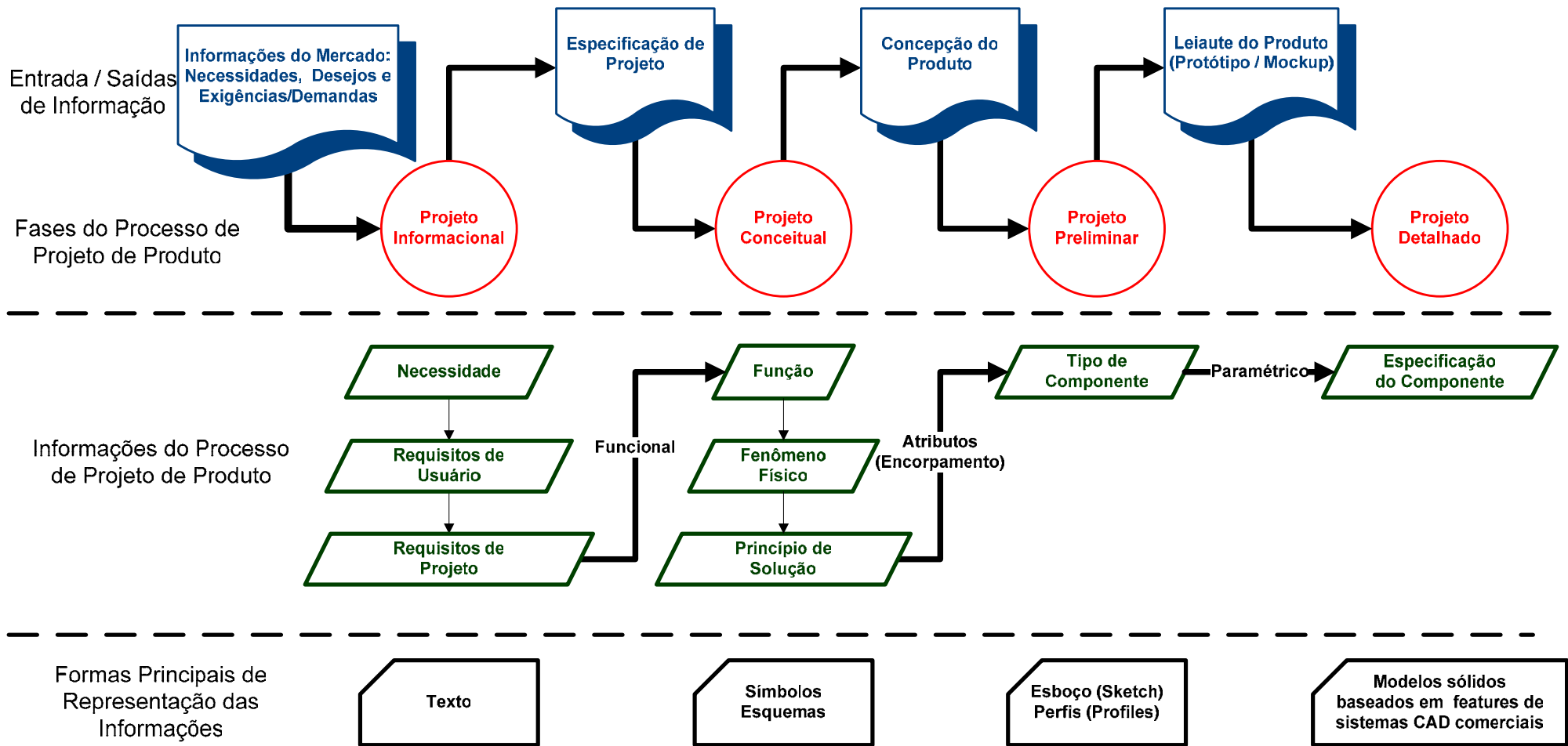


Figura 27. Fluxo de informações e a representação usada nas diferentes fases do PPP.

5.4. IDENTIFICAÇÃO E CLASSIFICAÇÃO DAS FEATURES DE PROJETO

5.4.1. CARACTERIZAÇÃO DAS FEATURES DE PROJETO

Para realização da tarefa de identificação e classificação das *features* de projeto deve-se ter em mente os seguintes aspectos, conforme sugerem (BRONSVOORT e JANSEN, 1993) e (SHAH e MÄNTYLÄ, 1995): o nível de abstração, o domínio da aplicação, e o tipo de produto.

O significado de uma informação sempre será dependente do (a):

- Conhecimento prévio e experiência sobre o assunto em discussão;
- Interpretação ou contexto do domínio de aplicação;
- Nível de abstração ou detalhamento do assunto;
- Forma de representação.

Nesse trabalho a princípio, esses aspectos se resumiram a duas questões básicas:

- Quais conceitos devem ser modelados para armazenar a evolução do processo de projeto de um produto?
- Quais informações podem ou devem ser armazenadas dentro de uma classificação de *features* de projeto?

O Capítulo 6 responde essas questões definindo as principais informações relacionadas às fases do processo de projeto de produto, e modelando-as como classes.

Abaixo, tem-se a listagem de algumas *features* que podem ser usadas durante a modelagem do projeto. As *features* da fase de projeto são referências a outras *features* que caracterizam a fase de projeto considerada.

Feature Informacional: questões do projetista, requisitos do projeto.

Atributos: *identificador, texto da questão, representação gráfica.*

Feature Conceitual: esboço, regiões de interesse ou de atenção especial no projeto, posicionamentos, restrições dimensionais e geométricas.

Atributos: *Dimensões de referência (comprimento, altura, largura) ainda*

imprecisas; Faixa de valores possíveis e/ou aceitáveis; Distâncias relativas; Forma ou perfil principal; Orientação.

Feature Preliminar: configurações de princípios de solução, leiaute intermediário.

Atributos: *nome, descrição, representação, função global, listas das features associadas.*

Feature Detalhada: diz respeito às *features* disponíveis nos sistemas CAD comerciais, num nível de detalhamento final do projeto. Os valores dos atributos são mais exatos, ou seja, com baixa probabilidade de incerteza. Isso se justifica até porque o projeto já está numa fase de detalhamento.

Atributos: *Dimensões nominais; tolerâncias, montagem, material, acabamento superficial, tratamentos térmicos e superficiais.*

Essas *features* foram definidas tomando por base o processo de projeto de produto, e verificando quais informações, o estágio de abstração, e a forma de representação delas na caracterização de cada fase. Essa visão ajuda a fazer um primeiro levantamento de quais informações já são possíveis de identificar e assim poder instanciá-las no projeto. Isso pode ser realizado na primeira fase de implementação da *featurização*, quando se faz o levantamento de informações sobre o produto.

5.4.2. CRITÉRIOS DE DEFINIÇÃO DOS ATRIBUTOS DAS FEATURES

Na classificação das *features* do processo de projeto, os atributos escolhidos para defini-las estão baseados em alguns critérios. Estes critérios usados são brevemente descritos na seqüência a seguir:

Critério 1: *Identificação da feature*

São atributos de identificação da *feature* dentro do sistema computacional, tais como o nome e o seu identificador. Os atributos dessa categoria são importantes principalmente para o sistema CAD, pois a partir deles pode-se saber quais *features* estão presentes no modelo de *features* em cada estado do projeto.

Critério 2: *Dados geométricos ou topológicos*

São atributos que informam valores de dimensões que caracterizam a *feature*, a forma, as tolerâncias.

Critério 3: *Dados ou informações da semântica da feature*

São atributos que guardam a informação da intenção do projetista, que possam ser utilizados tanto numa fase posterior do projeto ou atividade subsequente do ciclo de vida do produto. Esse tipo de informação deve ter uma natureza dinâmica, de adaptação ao contexto de aplicação do dado, caracterizando um fenômeno de polimorfismo.

5.4.3. A HIERARQUIA DE FEATURES

As *features* são modeladas como objetos que encapsulam os dados geométricos e topológicos, e a sua semântica (*identidade, estado, comportamento*) com atributos próprios de cada fase do projeto, os quais estão listados e comentados no Capítulo 6.

Numa primeira modelagem da hierarquia de *features* das fases de projeto, obteve-se a hierarquia de classes conforme mostrada na Figura 28.

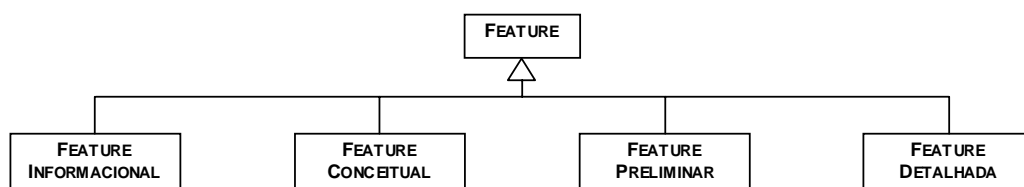


Figura 28. Primeira proposta de modelagem das *features* de fase do PPP.

A principal alteração na Figura 28, ocorreu no relacionamento das classes das *features* de fase do PPP. Na segunda modelagem das *features* de fases do PPP, as mesmas foram modeladas como classes *wrapper*². A classificação final das *features* com base no processo de projeto, que enfatiza a evolução do projeto durante as fases do PPP, é dada na Figura 29. A estrutura hierárquica assume uma característica de relacionamento generalização/especialização, através do conceito de herança de orientação a objeto.

² Classes *wrapper* são classes que definem todos os métodos de uma interface, mas apenas sobrecarregam os métodos de interesse, evitando assim a necessidade de implementar todos os métodos da interface.

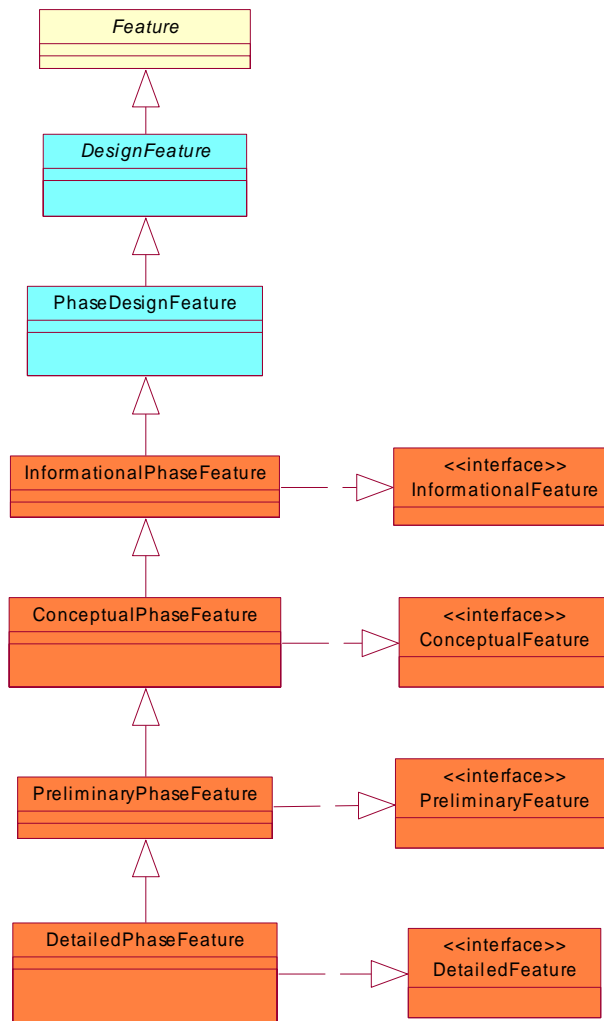


Figura 29. Hierarquia de *features* no processo de projeto.

A classe **FEATURE** é abstrata, ou seja, não existe instanciação de objetos desta classe. Em princípio, ela deve ter atributos genéricos para identificação da *feature*, e métodos de manipulação, tal como a instanciação, inserção e remoção da *feature* no modelo, etc.. A intenção é que outras classificações de *features*, em níveis mais inferiores da hierarquia de classes, possam herdar os seus atributos, e assim favorecer a reusabilidade dos dados.

As demais classes de *features* – **INFORMACIONAL**, **CONCEITUAL**, **PRELIMINAR**, e **DETALHADA** – são especializações da classe-pai **FEATURE**. Elas serão referências a outras *features*, de forma que representam a fase de projeto correspondente. Visto por esse ângulo, as *features* serão consideradas como uma “**identidade da fase de projeto**”, sendo representativas do estado atual de informação em cada fase de projeto.

A Figura 30 ilustra uma hierarquia mais geral de *features*, incluindo outros domínios de contexto de aplicação das *features*. Essa figura evidencia como o conceito de *features* é uma escolha interessante para modelagem de informações de Engenharia como um todo, constituindo-se numa base unificada de estruturação dessas informações. Ela também enquadra as *features* de fase de projeto com relação às demais classificações de *features*.

5.5. CONSIDERAÇÕES

É importante salientar que a representação dos dados e informações de projeto é dependente da modelagem realizada para esses dados. A construção de quaisquer outras unidades de informação, para uma reusabilidade em fases subseqüentes do processo de projeto ou do ciclo de vida de um produto, também é dependente da modelagem das classes mais genéricas, as quais servirão de base para a composição ou herança em *features* mais evoluídas, menos genéricas e abstratas, do processo de projeto.

Por isso que uma atenção especial deve ser dada à modelagem dos dados de projeto, que por sua vez vai decidir o sucesso ou não de uma proposta para troca de dados eficiente entre as diferentes fases de projeto. A estrutura de dados deve também servir para tornar mais eficiente a atividade de projetar o produto utilizando ferramentas computacionais apropriadas, tais como sistemas CAD otimizados e integrados com ferramentas e recursos de banco de dados.

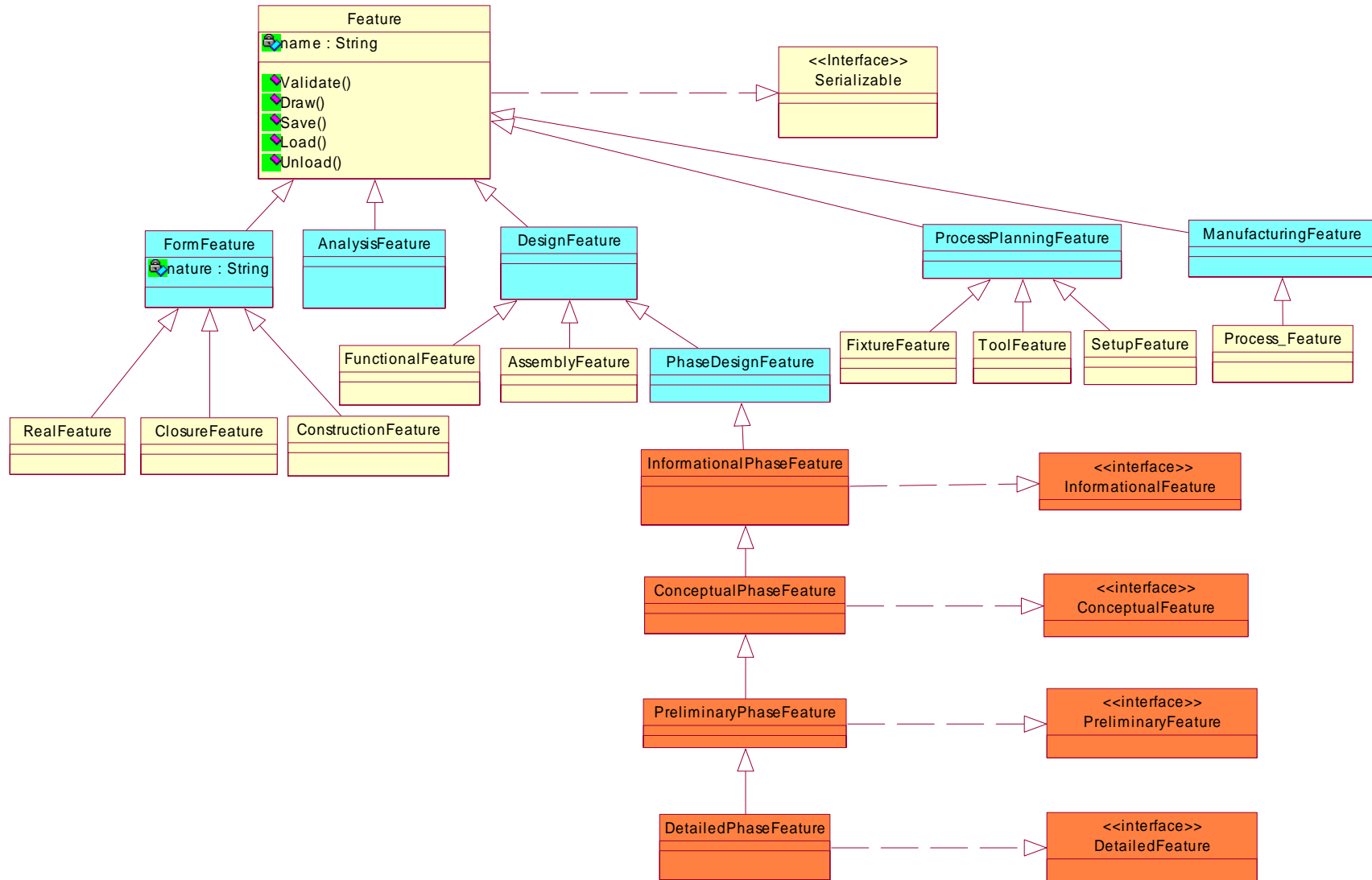


Figura 30. Hierarquia de *features*.

Capítulo 6

Resultados: Implementação dos Objetos e Features de Fase de Projeto

6.1. INTRODUÇÃO

A leitura de artigos, livros e teses principais nas áreas de Sistemáticas de Projeto, Desenvolvimento de Produto, e Sistemas CAD/CAE/CAM, permitiu comparar, verificar dubiedades, e decidir qual terminologia seria mais adequada levando-se em conta o critério contexto de projeto. O objetivo é selecionar, entre conceitos e terminologias empregadas nas bibliografias de referência, as informações referentes ao processo de projeto, e mais especificamente, às suas fases principais conforme citadas nos capítulos anteriores da tese.

Os objetos de projeto foram obtidos a partir da seleção entre vários outros substantivos extraídos de textos da área de pesquisa em processo de projeto, desenvolvimento de produto e sistemas CAD/CAM. Nessa tese, foram selecionadas as terminologias, conceitos e definições encontrados e extraídos dos textos relacionados ao processo de projeto de autoria de

pesquisadores da área, os quais compreendem a maioria das referências bibliográficas listadas. O critério de seleção dos nomes foi feito com base no nome eleito ter um significado e uma semântica definida para o contexto de fases do processo de projeto.

Após uma sessão inicial de eleição dos objetos que participam do contexto de processo de projeto foram identificados 32 (*trinta e dois*) objetos. A Figura 31 ilustra as classes que implementam esses objetos.

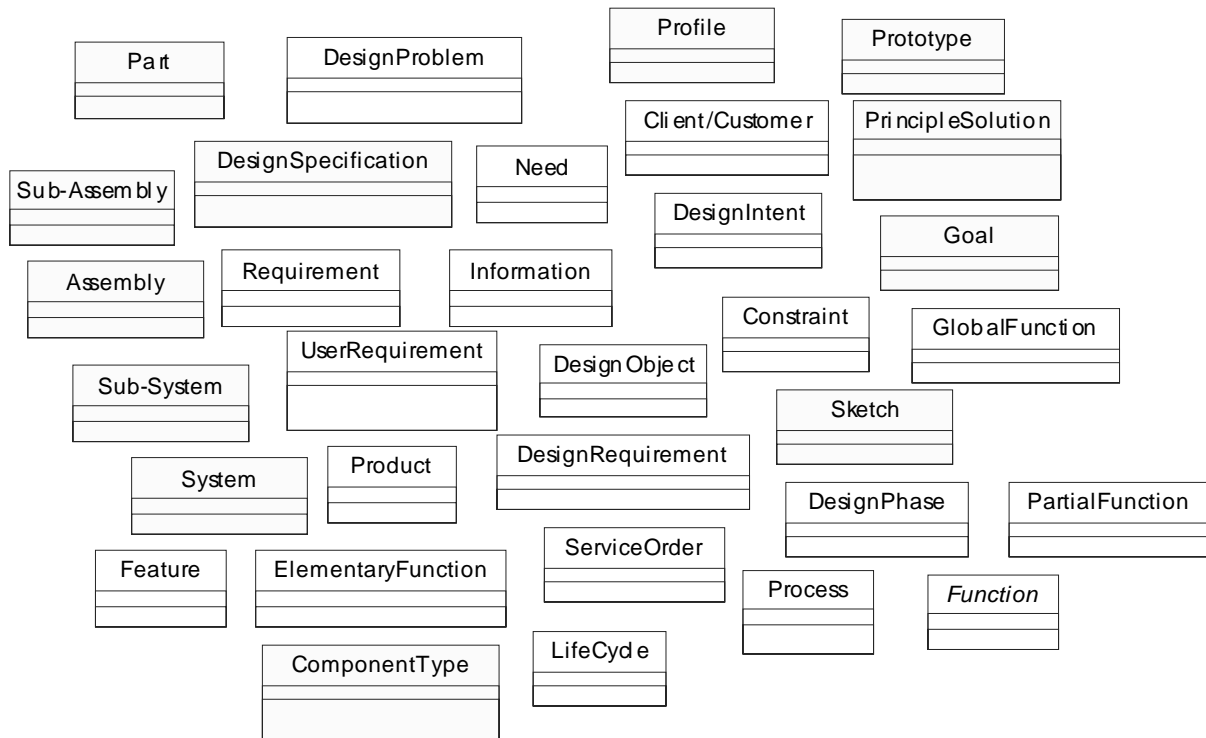


Figura 31. Classes do contexto do PPP.

Esse trabalho de tese propõe um modelo do PPP numa visão holística do mesmo com base, principalmente, nos artigos de (DIXON *et al.*, 1988), (BRUNETTI e GOLOB, 2000) e (FONSECA, 2000). Essa modelagem, ilustrada pela Figura 32, é caracterizada por relacionamentos do tipo todo-parte como composições/agregações das classes do contexto do processo de projeto.

Na modelagem, o processo de projeto é visualizado como constituído por três outras classes, a saber: a classe de problema de projeto (**DesignProblem**), a classe projetista (**Designer**) em si; e a classe do contexto (**Context**) que o encerra. O problema de projeto, por sua vez, fica constituído por duas outras classes, a de estado inicial do conhecimento e a de estado final do conhecimento, conforme descrito em (DIXON *et al.*, 1988).

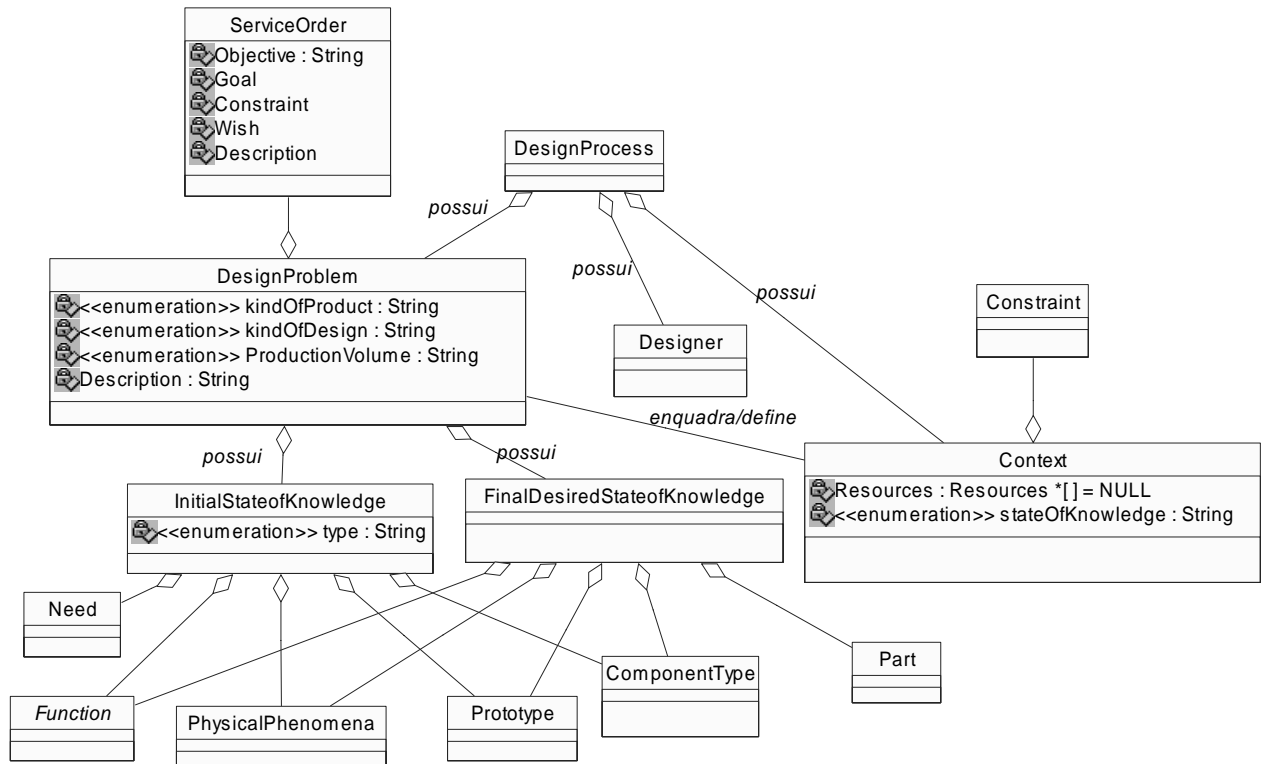


Figura 32. Diagrama UML de classes do PPP.

6.2. FEATURES COMO UNIDADE DE INFORMAÇÃO PARA OBJETOS DE PROJETO

Os objetos identificados nesse texto, de início, estão modelados com uma visão que tenta explicitar atributos e métodos para a definição dos mesmos dentro do contexto do processo de projeto mecânico. Esses são os tipos de atributos e métodos mais importantes para essa fase do trabalho, que é identificar e definir bem a semântica do objeto, ou seja, definir os atributos que possuem alguma relação com o processo de projeto. Por exemplo, no caso de um objeto da semântica de necessidade, atributos de tipificação da necessidade são essenciais para a utilização dentro do processo de projeto, pois uma necessidade tipo qualificada como exigência é muito mais prioritária do que se ela é apenas um desejo. Ter um atributo assim na classe necessidade é muito útil para o processo de projeto. Pode-se acrescentar que nessa fase se deseja explicitar as propriedades e atributos intrínsecos ao objeto.

Outros aspectos também importantes – tais como atributos e métodos informativos, ou que permitam gerenciar, validar e identificar o objeto; estão sendo deixados para uma modelagem posterior. Exemplos desses atributos podem ser tais como o identificador (ID) do objeto, o qual apenas o unifica e permite assim diferenciá-lo dos outros. Esses atributos serão importantes tanto para a implementação de protótipos computacionais que utilizem essas classes, e também possibilitar a geração de informação útil, na forma de relatórios ou arquivos “log”, para o suporte ao processo de projeto.

Na seqüência das seções 6.2.1 até 6.2.3, foram definidas e descritas as classes que implementam as *features* das fases do processo de projeto. Em cada seção, foram evidenciadas as *features* que compõem cada fase, os seus atributos e citados alguns métodos. Ao final de cada seção, é ilustrada uma figura representando o diagrama de classes UML das *features* de fase, evidenciando os correspondentes relacionamentos entre essas *features*. Ao longo do texto dessas seções, também são citadas as referências com os respectivos comentários que nos ajudaram a argumentar na escolha e/ou definição da *feature*.

6.2.1. CLASSES DA FASE DE PROJETO INFORMACIONAL

- **CLASSE NECESSIDADE (*NEED*)**

ATRIBUTOS

- **Identificador (ID) / Nome:** nome que identifica a necessidade;
- **Descrição/Comentário:** texto que descreve e/ou comenta a necessidade nas palavras do cliente;
- **Coefficiente de prioridade (CP):** indica o nível de importância dessa informação no projeto nas palavras do cliente;
- **Classe:** tipo de categoria a qual a necessidade se refere. Esse atributo é baseado nas fases que compõem o ciclo de vida do produto. Como sugestão, pode-se utilizar informações do modelo da espiral do desenvolvimento do produto apresentado em (FONSECA, 2000);
- **Tipo** {Exigência/Demanda, Desejo}: atributo que classifica a necessidade como desejo ou exigência/demanda.

MÉTODOS

Todos os métodos de definição (SET) e de consulta (GET) para os atributos da classe.

Ex.: Definir o coeficiente de prioridade (CP); Consultar o tipo da necessidade.

- **CLASSE REQUISITO DE USUÁRIO (*USER REQUIREMENT*)**

A lista de requisitos resulta das primeiras iterações, ainda na fase de clarificação das idéias (*Projeto Informacional*), e descreve as restrições gerais que um produto tem ou deve satisfazer.

Os requisitos de usuário são criados com base nas necessidades do cliente, na tecnologia disponível, e na estratégia de marketing de uma companhia/corporação/empresa. Eles podem ser usados para especificar funções do produto a ser desenvolvido, e neste quase o qualifica como um indutor funcional.

ATRIBUTOS

- **Identificador (ID) / Nome:** nome que identifica o requisito;
- **Descrição/Comentário:** texto que descreve e/ou comenta o requisito de usuário;
- **Tipo** {Demanda/Exigência, Desejo, Indutor funcional};
- **Classe** {Funcionalidade, Manufaturabilidade / Fabricabilidade, Economia, Usuário / Ambiente}; Classificação pelo tipo de aplicação;
- **Natureza:** sub-classificação associada à natureza da classe do requisito;

Ex.: Sub-classe {geométrica, cinemática, energia, material, sinal, ...}

- **Característica** {Qualitativa, Quantitativa};

- **CLASSE REQUISITO DE PROJETO (*DESIGN REQUIREMENT*)**

É importante comentar que o conceito de requisito é entendido na literatura técnica (BRUNETTI e GOLOB, 2000; FONSECA, 2000; LEE e LEE, 2001; VAN VLIET *et al.*, 2001) como possuindo uma evolução em termos de definição da informação que ele deseja passar em diferentes momentos do processo de projeto. Uma diferença entre um requisito de projeto e um

requisito de usuário, é que o requisito de usuário é associado a um substantivo, e que o requisito de projeto segue uma linguagem técnica e refere-se a uma grandeza da peça/produto, possuindo um valor associado, ou seja, ele é necessariamente quantificado.

Na fase de projeto informacional, o requisito surge como requisito de usuário e depois evolui para o requisito de projeto. No modelo de classes UML da fase de projeto informacional, essa evolução é implementada como um relacionamento de herança, onde a classe **DesignRequirement** herda os atributos da classe **UserRequirement**.

ATRIBUTOS

- **Propriedades:** atributo que qualifica a grandeza que se quer valorar no requisito de projeto, a qual pode ser física, química, elétrica, etc..
Ex.: Geométricas {tamanho, altura, largura, comprimento}
- **Valor:** numeral associado à grandeza do requisito de projeto;
- **Unidade dimensional:** atributo que define a unidade dimensional da grandeza;

Utilizando as duas classes de requisitos modeladas acima para exemplificar como a modelagem orientada a objetos favorece a organização das informações de projeto, pode-se comentar o seguinte: Os atributos de classe e natureza do requisito de usuário podem servir como informação prévia para métodos da classe requisito de projeto sugerir ao usuário da classe propriedades relevantes associadas. Considere um requisito de usuário, cujos atributos **classe** e **natureza** tenham valores *ambiente* e *energia*, respectivamente. Esses atributos podem sinalizar ao requisito de projeto associado faixas de níveis aceitáveis de consumo/desempenho energético do produto que está sendo projetado.

- **CLASSE ESPECIFICAÇÃO DE PROJETO (*DESIGN SPECIFICATION*)**

O objeto principal resultante da fase de projeto informacional pode ser representado pelas especificações de projeto.

As especificações de projeto são constituídas pelos *requisitos de projeto* com suas metas, objetivos e restrições individuais, indispensáveis para o desenvolvimento do projeto, mais as metas, objetivos e restrições do projeto como um todo.

A diferença entre requisito de projeto e especificação de projeto está na definição ou não das metas, objetivos e restrições. Nas especificações de projeto, estes precisam ser definidos, mas nos requisitos de projeto não precisam. Um requisito de projeto passa a ser uma especificação de projeto, quando tem definido suas próprias metas, objetivos e restrições.

Para reforçar a definição e identificação de atributos e métodos para a especificação de projeto, segundo (VAN VLIET *et al.*, 2001) define-se que o resultado da fase de especificação do projeto (Projeto Informacional) é a especificação do projeto do produto (**PDS** – **Product Design Specification**), freqüentemente referida como **lista de requisitos**.

A lista de requisitos contém:

- Condições de contorno, os quais definem as interfaces do produto com o ambiente do produto. Pode ser uma informação quantificada ou apenas qualificada.
- Demanda ou Necessidade, a qual define a funcionalidade ou propriedades que o produto deve ter.
- Restrições, as quais definem os limites quantificados do espaço de solução para as várias propriedades do produto.
- Desejos, os quais descrevem a funcionalidade ou propriedades do produto que não são estritamente requeridos, mas desejados para melhorar a performance do produto.

(VAN VLIET *et al.*, 2001) estabelecem importantes requisitos que a **PDS** deve atender/satisfazer:

- Ela deve conter um método para ordenar desejos, de forma a ser possível avaliar concepções de projeto;
- Ela deve ser um documento dinâmico. Isso significa ser um documento capaz a evoluir e alterar-se através de todo o processo de projeto.
- Ela deve possibilitar tratar efetivamente com o caráter incompleto, indefinido e incerto da informação.
- Ela deve habilitar o efetivo reuso da informação e também da intenção de projeto ser armazenada.

As definições de metas, objetivos e restrições do projeto, citadas na

seqüência, são as mesmas definidas com base no trabalho de tese de (FONSECA, 2000).

As metas são os valores a serem definidos nas métricas associadas aos requisitos de projeto selecionados. Podem estar contidas no problema de projeto, ou na maioria das vezes, devem ser definidas pelos projetistas, como por exemplo: *“cor ofuscante e verde”* ou *“peso igual ou menor do que dois quilogramas”*.

Os objetivos são definições importantes contidas no problema de projeto, ou a serem levantados pela equipe de projeto, que expressam o que se procura com determinado requisito ou trabalho de projeto, como por exemplo: *“a cor deve estimular a atenção dos usuários”* ou *“o produto tem que ser portátil”*.

As restrições são limitações, impostas no problema de projeto e decididas pela equipe de projeto ou impostas por normas, tais como: *“não usar material metálico”* ou *“nunca maior que cinco quilogramas”*.

ATRIBUTOS

- **Lista dos requisitos de projeto:** atributo que armazena todos os requisitos de projeto com suas respectivas metas, objetivos e limitações preenchidos;
- **Meta do projeto:** atributo que descreve as metas do projeto como um todo;
- **Objetivo do projeto:** atributo que descreve os objetivos do projeto como um todo;
- **Restrições/Limitações do projeto:** atributo que descreve as limitações do projeto como um todo;

A Figura 33 mostra o diagrama UML das classes que compõem à fase de projeto informacional. Nesse diagrama ilustra os relacionamentos entre as classes, com ênfase para as heranças de requisito de projeto herda requisito de usuário, que por sua vez herda necessidade. E a composição de requisito de projeto compondo as especificações de projeto. Importante também ressaltar a herança de função herda requisito de usuário nos casos que o seu atributo “str_type” for qualificado como um valor de indutor funcional.

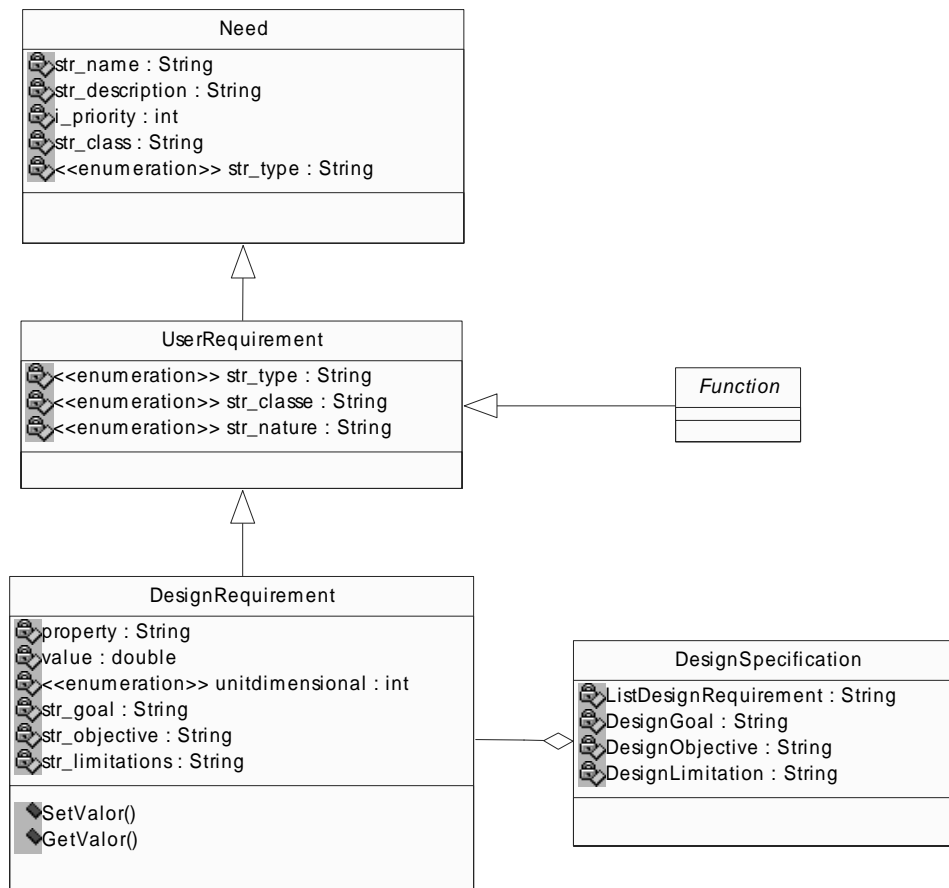


Figura 33. Diagrama UML de classes do projeto informacional.

- **CLASSE FEATURE DA FASE INFORMACIONAL (INFORMATIONAL PHASE FEATURE)**

A classe **InformationalPhaseFeature** de *feature* informacional foi modelada como uma classe do tipo *wrapper*, implementando a interface **InformationalFeature**, conforme destacado na Figura 29 e Figura 30 do Capítulo 5.

Sendo assim, os objetos de projeto das classes informacionais que herdam **InformationalPhaseFeature** devem declarar somente os métodos de interesse desta classe, e sobrecarregá-los quando necessário.

ATRIBUTOS

- **Quantidade:** armazena a quantidade de *features* informacionais instanciadas;
- **Indicador:** atributo ponteiro que aponta para uma *feature* informacional;

MÉTODOS

- Classificar necessidades por um critério; por exemplo, pelo seu tipo e/ou classe;
- Ordenar necessidades por um critério; por exemplo, pelo seu coeficiente de prioridade;
- Converter de necessidades em requisitos de usuário.
- Converter de requisitos de usuário indutor funcional em função do produto;
- Ordenar os requisitos de projeto com base num critério;

6.2.2. CLASSES DA FASE DE PROJETO CONCEITUAL

- **CLASSE FUNÇÃO (*FUNCTION*)**

Em sistemas técnicos, a função do produto envolve alguma conversão/transformação de elementos básicos, tais como: energia, material e/ou sinal (*informação*). Tais elementos são denominados de fluxo.

A classe função é abstrata, ou seja, ela não instancia objetos. Todavia, outras classes podem herdar seus atributos e métodos e implementá-los, a exemplo do que acontece com as classes de funções global, parcial e elementar, pois as mesmas são classes do tipo função.

ATRIBUTOS

- **Identificador (ID) / Nome**
- **Tipo** {Global, Parcial, Elementar}: atributo que especifica de qual tipo é a função;
- **Intenção**: atributo que especifica o propósito da função;

- **CLASSE FLUXO (*FLOW*)**

ATRIBUTOS

- **Tipo do fluxo** {Energia, Material, Sinal}: atributo que especifica o tipo de fluxo processado pela função;
- **Orientação** {Entrada, Saída}: atributo que indica se o fluxo é de entrada ou de saída.
- **Propriedades**: atributo que especifica o tipo da grandeza associada ao fluxo processado pela função.

Por exemplo: Se o fluxo for de energia, tem-se mecânica, térmica, elétrica, magnética, acústica, ótica, química, etc..

- **Valor:** numeral associado à propriedade com a respectiva unidade;
- **Unidade dimensional:** atributo que define a unidade dimensional da grandeza;
- **CLASSE ESTRUTURA FUNCIONAL (*FUNCTIONAL STRUCTURE*)**

ATRIBUTOS

- **Função principal:** atributo que especifica o identificador da função que encabeça a estrutura funcional;
- **Lista das sub-funções:** atributo que armazena todas as sub-funções (*parciais e elementares*) que compõem a estrutura funcional.
- **Lista dos fluxos:** atributo que armazena os fluxos de cada função da estrutura funcional;
- **CLASSE PRINCÍPIO DE SOLUÇÃO (*PRINCIPLE OF SOLUTION*)**

ATRIBUTOS

- **Tipo de efeito:** atributo que especifica o tipo de efeito associado a uma lei física que representa o efeito. Por exemplo: efeito de atrito,
- **Portador de efeito:** atributo que especifica o elemento que possibilita a atuação do efeito;
- **CLASSE MÓDULO REALIZÁVEL (*FEASIBLE MODULE*)**

ATRIBUTOS

- Lista de princípios de solução;
- Lista de funções;

A Figura 34 mostra o diagrama UML das classes que compõem à fase de projeto conceitual.

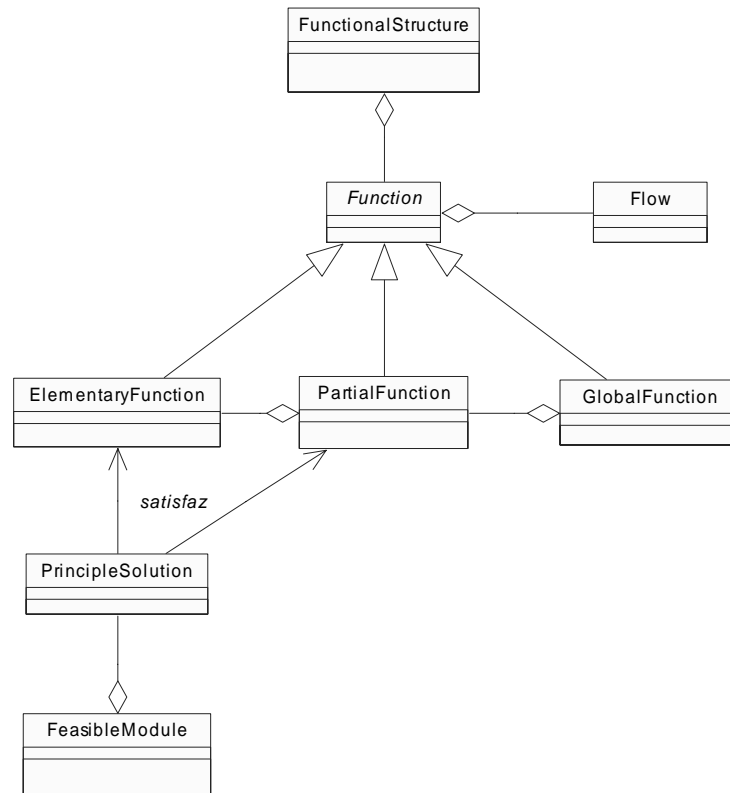


Figura 34. Diagrama UML de classes do projeto conceitual.

O diagrama da Figura 34 ilustra os relacionamentos entre as classes, com ênfase para as heranças das classes de função **ElementaryFunction**, **PartialFunction** e **GlobalFunction** a partir da classe abstrata **Function**, e as composições entre as classes de **PrincipleSolution** compondo **FeasibleModule**, **Function** compondo **FunctionalStructure**, e **Flow** compondo **Function**.

- **CLASSE FEATURE DA FASE CONCEITUAL (CONCEPTUAL PHASE FEATURE)**

ATRIBUTOS

- **Quantidade:** armazena a quantidade de *features* conceituais instanciadas;
- **Indicador:** atributo ponteiro que aponta para uma *feature* conceitual;

MÉTODOS

Métodos associados as principais técnicas aplicadas no projeto conceitual:

- Métodos para classificar soluções alternativas por um critério pré-estabelecido.
- Método para compor a estrutura funcional do produto;
- Método para compor a estrutura de princípios de solução e módulos realizáveis.

6.2.3. CLASSES DA FASE DE PROJETO PRELIMINAR E DETALHADO

Na identificação das classes do projeto detalhado, a modelagem fez uma diferenciação entre peça e produto com base em artigos referenciados no texto.

Segundo (SRINIVASAN, 1999), um produto é recursivamente definido como uma montagem de subsistemas e peças. Peças, por sua vez, são constituídas por *features*, as quais estão sujeitas as várias condições (*restrições*), como é mostrado na Figura 35.

É importante fazer um conjunto de observações sobre essa taxonomia. Primeiramente, como observado acima, a definição recursiva de um produto termina em peças. Está comprovado que é complicado ter uma definição aceitável e consensual de peças.

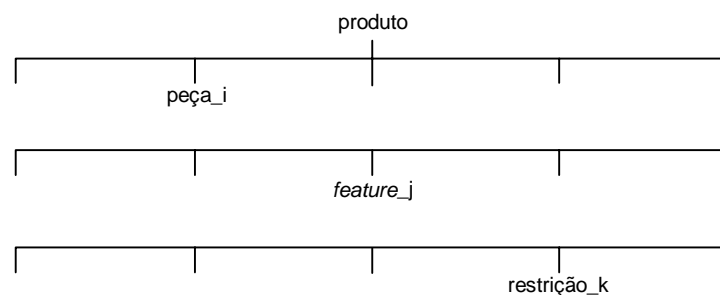


Figura 35. Especificação da geometria de um produto.

Uma visão científica tenta definir uma peça como algo que pode ser modelado como um sólido rígido, isto é, através de um conjunto de operações regulares de modelagem de sólido.

Por outro lado, uma visão da indústria, define uma peça como algo que possui uma numeração de peça no produto. Esta é uma visão prática porque ela está mais de acordo com a forma como uma indústria ou organização de Engenharia identifica um produto.

Essa diferença de visão/definição entre a indústria e a comunidade científica, sobre o que é uma peça, é melhor notada através de um exemplo. Considere que um projetista de um computador PC pode visualizar uma unidade de disquete (*drive*) como uma única peça, e dar a ele uma numeração de peça. Por sua vez, um projetista de unidades de disquete pode considerar um rolamento que pertence à unidade de disquete como uma peça, e dar a ele uma numeração de peça.

Em ambos os casos descritos acima, o que os projetistas identificaram como uma peça, não podem ser consideradas peças pela definição científica, pois eles não são corpos únicos rígidos, mas uma montagem constituída por vários sólidos.

Conclui-se que a dificuldade de se conseguir uma boa definição de peça para fins de recursão da definição do produto não é um problema, a não ser que a especificação geométrica do produto seja também considerada. O ponto-chave em superar esta dificuldade é focar sobre o conceito de *features*, e verificar se o relacionamento entre *features* pode ser definido independente se as *features* pertencem à mesma peça ou a diferentes peças.

A conclusão importante é que os padrões ou normas devem focar exclusivamente sobre *features* e sobre restrições impostas sobre estas *features*. Essa diretriz pode ser depois usada mais adiante para especificar peças ou produtos quando necessário.

Uma segunda conclusão é que as restrições sobre uma *feature* são de dois tipos: intrínsecas ou relacionais. Exemplos de restrições intrínsecas são: o tamanho e a forma, pois estão definidas sobre a *feature*. E um exemplo de restrição relacional é a posição de uma *feature*, pois é definida em relação à outra *feature*.

Interessante observar que aqui, o termo **restrição** é usado com o mesmo significado que o termo **propriedade** é usado em (SHAH e MÄNTYLÄ, 1995). Assim, têm-se as propriedades divididas em dois tipos: intrínsecas e extrínsecas; ou seja, a mesma classificação que os tipos de restrição citados acima, respectivamente.

A Figura 36 mostra as classes do projeto detalhado, evidenciado os relacionamentos de herança e composição entre as classes dessa fase.

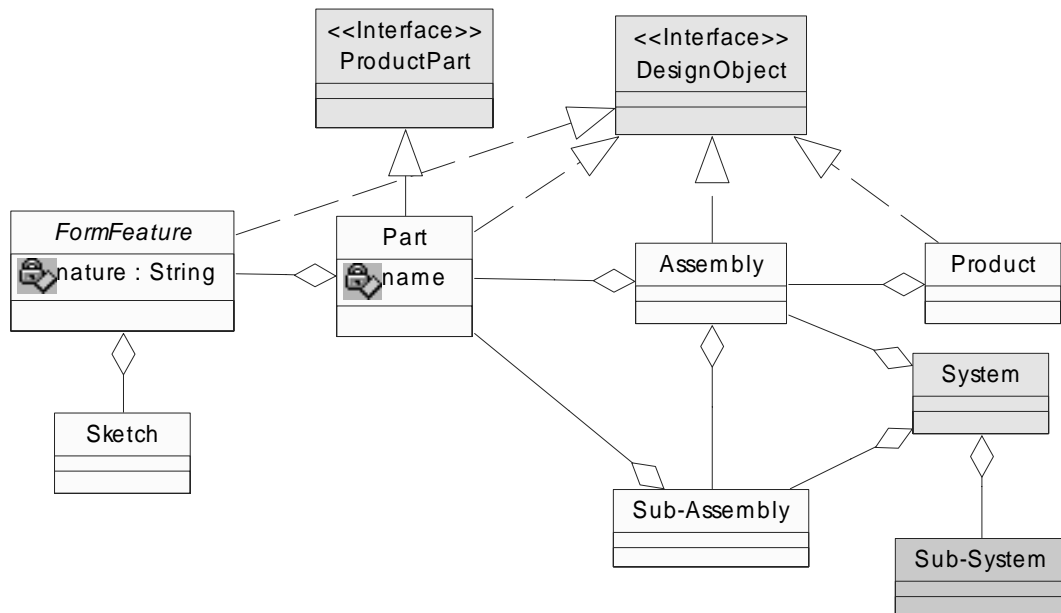


Figura 36. Diagrama UML de classes do projeto preliminar e detalhado.

- **CLASSE FEATURE DA FASE DETALHADA (*DETAILED PHASE FEATURE*)**

ATRIBUTOS

- **Quantidade:** armazena a quantidade de *features* detalhadas instanciadas;
- **Indicador:** atributo ponteiro que aponta para uma *feature* detalhada;

MÉTODOS

- Métodos para definição dos atributos finais do projeto.

6.3. ESTUDO DE CASO

Nessa seção, será mostrado o estudo de caso desenvolvido com o objetivo de exemplificar a utilização no processo de projeto das classes modeladas.

O estudo de caso relata o projeto de um aparelho portátil de hidromassagem (FERRÃO e SEGATTO, 2001), enfatizando os processos metodológicos dentro de cada fase do processo de projeto e a utilização das classes de *features* de fase na captura e armazenamento dessas informações.

6.3.1. EXEMPLIFICANDO NO PROJETO INFORMACIONAL

- **Necessidades do usuário:**

“Necessita-se de um aparelho **portátil** que **produza jatos** de água a uma determinada **vazão e temperatura, conforme regulagem** feita pelo usuário. Este aparelho deve **ser portátil** para poder ser levado junto com o usuário para lugares como piscinas, banheiras e boxes de banheiro. Por fim, o aparelho **deve ter um preço acessível e deve ser seguro e durável.**”

A partir da declaração das necessidades, pode-se utilizar a classe **Need**, fazendo a instanciação de objetos dessa classe, para armazenar as necessidades. Por exemplo:

Objeto **Need01**:

Atributo Identificador (ID): Need01

Atributo Descrição/Comentário: necessita-se de um aparelho portátil;

Atributo Coeficiente de prioridade (CP): 3 (*valor inteiro*);

Atributo Classe: Mobilidade;

Atributo Tipo: Exigência/Demanda

- **Requisitos do usuário:**

Ser portátil;

Leve;

Produzir jato / Ser pressurizada;

Ter preço acessível / Ser barato;

Permitir regulagem de temperatura e pressão;

Ser seguro;

Ter durabilidade / Ser durável

Os atributos do objeto **Need01** instanciado acima, pode ser herdado ou sobrecarregados por um objeto da classe **UserRequirement**:

Objeto **UserRequirement01**:

Atributo Identificador (ID): UserRequirement01;

Atributo Descrição/Comentário: necessita-se de um aparelho portátil;

Atributo Coeficiente de prioridade (CP): 3 (*valor inteiro*);

Atributo Tipo: Exigência/Demanda;

Atributo Classe: Funcionalidade;

Atributo Natureza: Geométrica

Atributo Característica: Quantitativa;

- **Requisitos do projeto:**

Ser portátil: implica **ter pequeno volume** (*grandeza volume*) e baixo peso (*grandeza peso*);

Permitir regulagem de **temperatura** e **pressão**: implica estabelecer uma faixa de valores para temperatura (*grandeza temperatura*) e pressão (*grandeza pressão*);

Ter **preço** acessível / Ser barato: implica baixo custo (*grandeza custo*);

Ser seguro: implica material **isolante** (*grandeza condutibilidade elétrica*);

Ter durabilidade / Ser durável: implica características do material do produto e da montagem das peças, como ser resistente a ambientes úmidos (*grandeza impermeabilidade, vedação do conjunto*) e ser resistente ao cloro.

Os atributos do objeto **UserRequirement01** instanciado acima, pode ser herdado ou sobrecarregados por um objeto da classe

DesignRequirement:

Objeto **DesignRequirement01:**

Atributo Propriedade: Volume

Atributo Valor: 15;

Atributo Unidade dimensional: litro (l);

- **Casa da qualidade (QFD – Quality Function Deployment):**

De acordo com a Casa da Qualidade, mostrada na Figura 37, os valores de importância dos requisitos de projeto e sua classificação fica sendo esta:

Funcionalidade;

Leve;

Segurança;

Compacto;

Regulável;

Esteticamente arredondado;

Ergonomia;

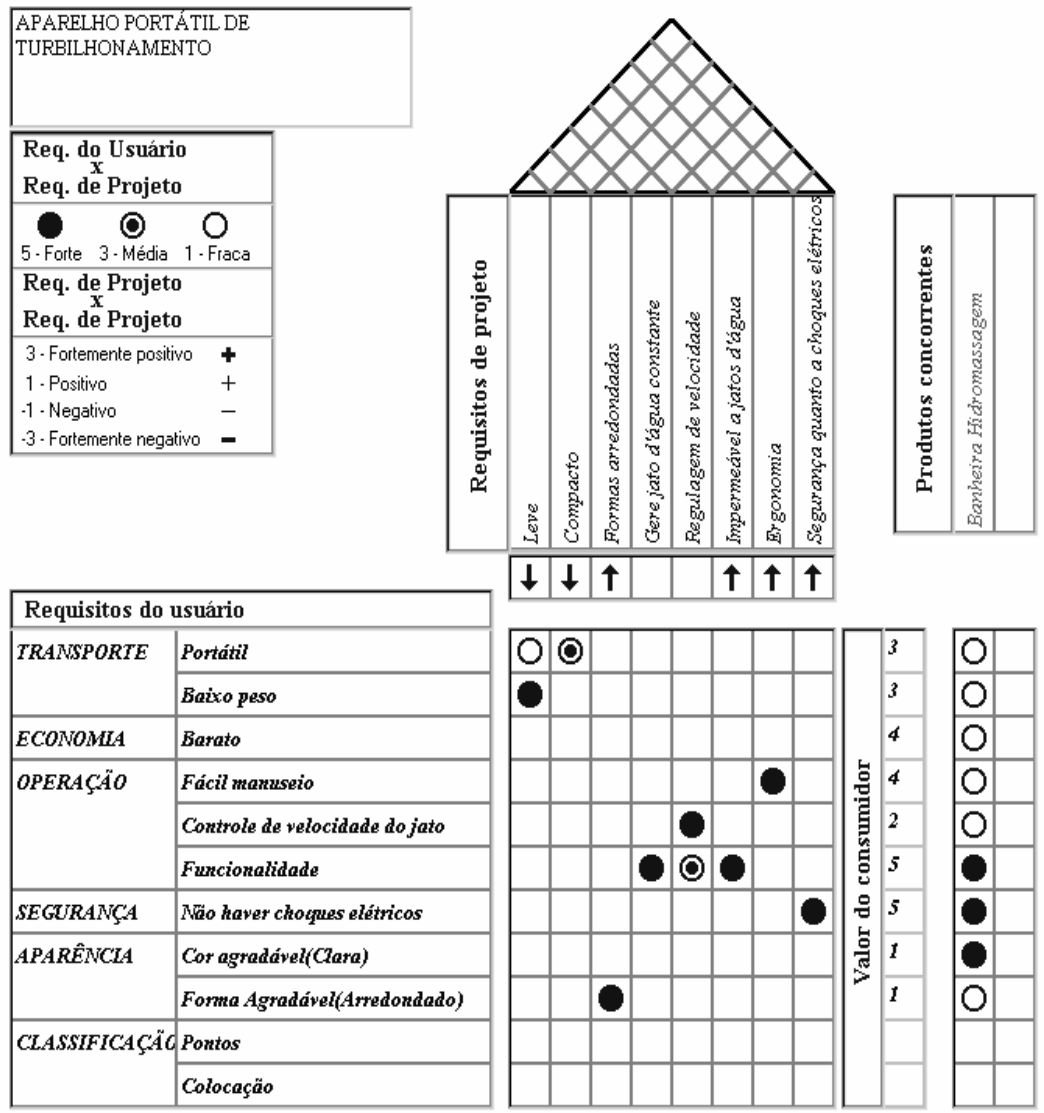


Figura 37. Casa da qualidade do aparelho portátil de hidromassagem.

6.3.2. EXEMPLIFICANDO NO PROJETO CONCEITUAL

No caso do Projeto Conceitual, tem-se como exemplo a estrutura funcional do produto, mostrada na Figura 38, a qual é capturada pela classe **FunctionalStructure**.

Todas as classes que constituem a estrutura funcional do produto foram modeladas com atributos para armazenar os seus dados pertinentes, e representar a relação hierárquica e de composição entre os tipos de função.

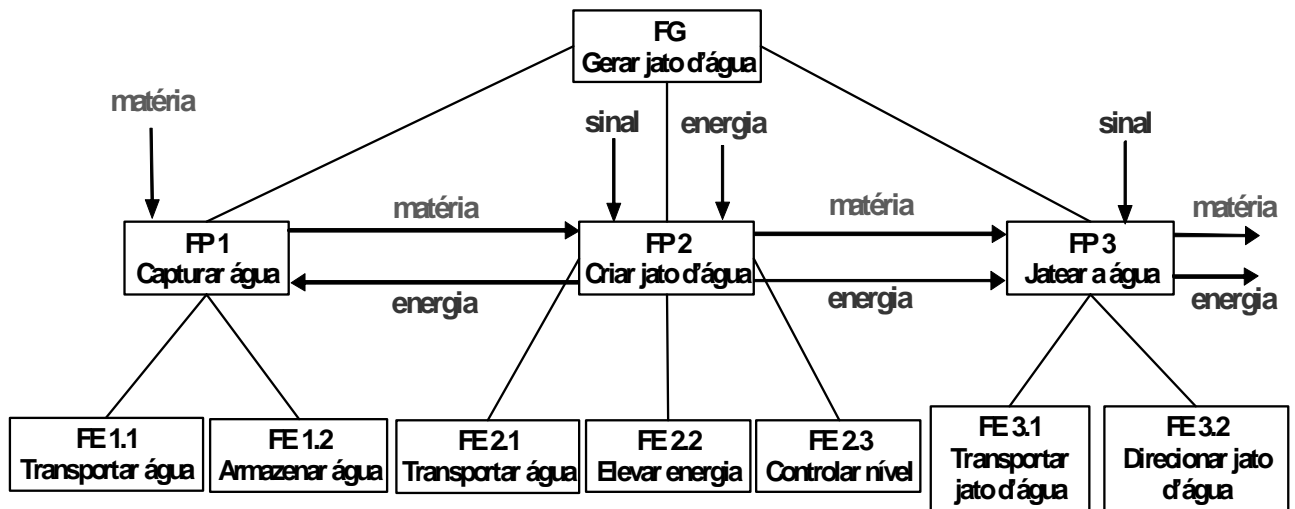


Figura 38. Estrutura funcional do aparelho portátil de hidromassagem.

Considere a instanciação de um objeto **ElementaryFunction** com os seguintes atributos:

Atributo Identificador (ID): FE2.2;

Atributo Tipo: Elementar;

Intenção: Elevar energia;

Esse objeto compõe um outro objeto que é uma função parcial da classe **PartialFunction**, cujos atributos são:

Atributo Identificador (ID): FP2;

Atributo Tipo: Parcial;

Intenção: Criar jato d'água;

O objeto **Flow** que compõe essa função parcial compreende os tipos sinal (entrando), e energia (entrando, saindo).

A partir desses objetos, pode-se comentar um fato importante e que constitui uma das vantagens da orientação a objeto que é o polimorfismo. Assim como existe a função parcial “Criar jato d'água”, que nos leva à função elementar “Elevar energia”, a qual nesse caso está associada a um fluxo de energia de pressão.

Uma outra estrutura funcional do aparelho portátil de hidromassagem, mostrada na Figura 39, associada a um fluxo de energia térmica por uma função parcial “Aumentar temperatura”, também tem uma função elementar associada “Elevar energia térmica”.

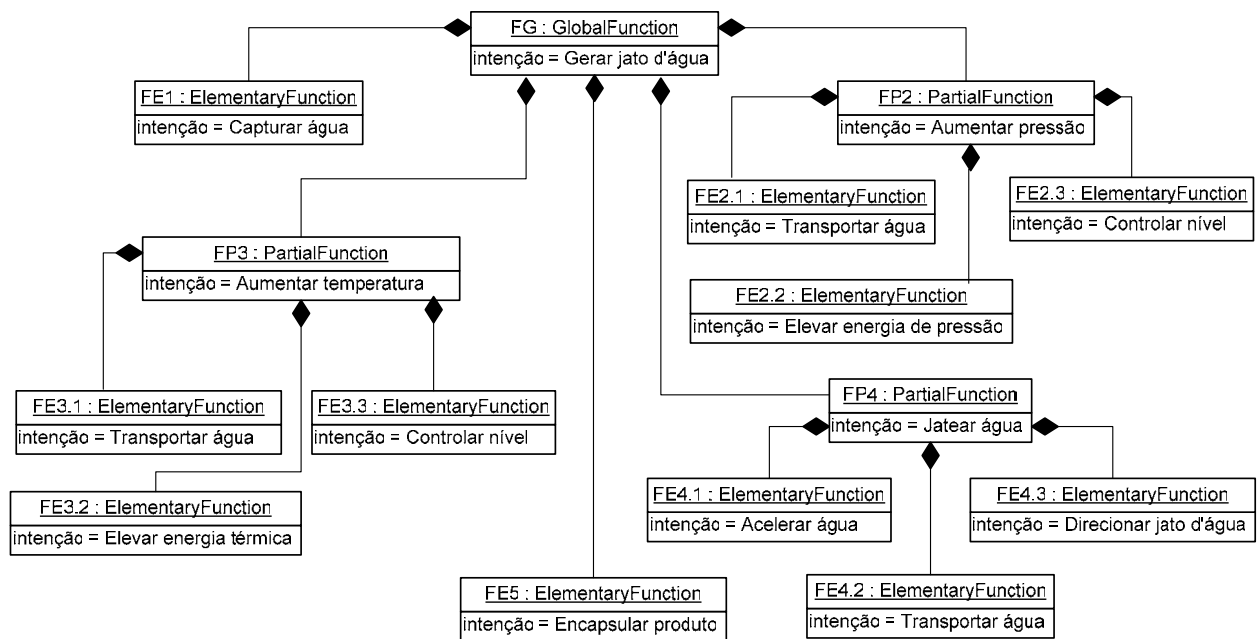


Figura 39. Estrutura funcional do aparelho portátil de hidromassagem em UML.

Através de um mesmo objeto **ElementaryFunction**, o atributo **Intenção** com valor “Elevar energia” pode ser associado a um fluxo de energia de pressão para satisfazer à função parcial “Criar jato d’água”, ou pode ser associado a um fluxo de energia térmica para satisfazer à função parcial “Aumentar temperatura”. O que vai determinar isso é a quem o objeto **ElementaryFunction** estiver associado.

6.3.3. EXEMPLIFICANDO NO PROJETO PRELIMINAR E DETALHADO

A utilização das classes das fases de projeto preliminar e detalhado é mostrada através da Figura 40, onde se tem um croqui do leiaute selecionado na fase de projeto conceitual para o sistema de aquecimento da água.

Um objeto da classe **Sketch** é instanciado para fazer referência a esse croqui, o qual possui várias peças do sistema de aquecimento.

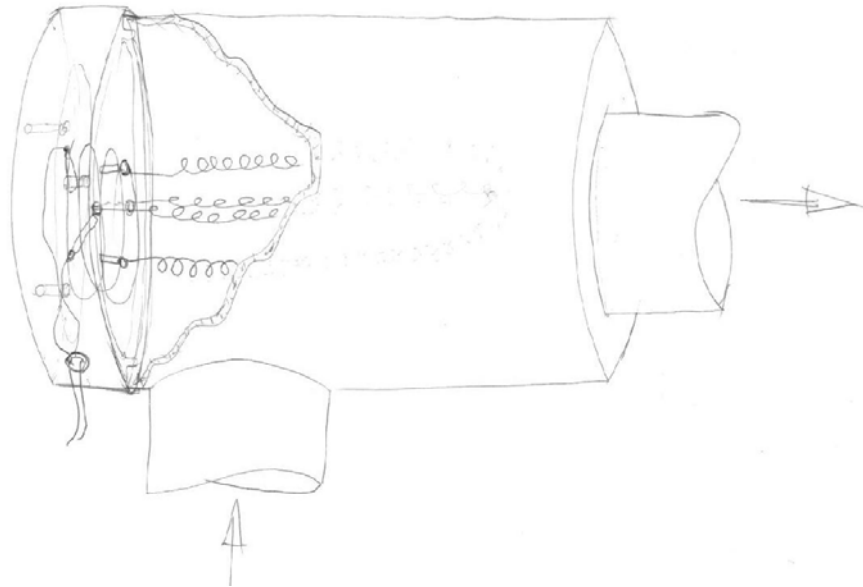


Figura 40. Croqui do leiaute do sistema de aquecimento da água do aparelho portátil de hidromassagem.

O projeto preliminar das peças não teve nenhum dimensionamento, somente determinaram quais deveriam ser as peças do sistema, suas quantidades, geometria, localização e funcionamento. Estas podem ser vistas no croqui da Figura 41, representando a vista explodida do sistema.

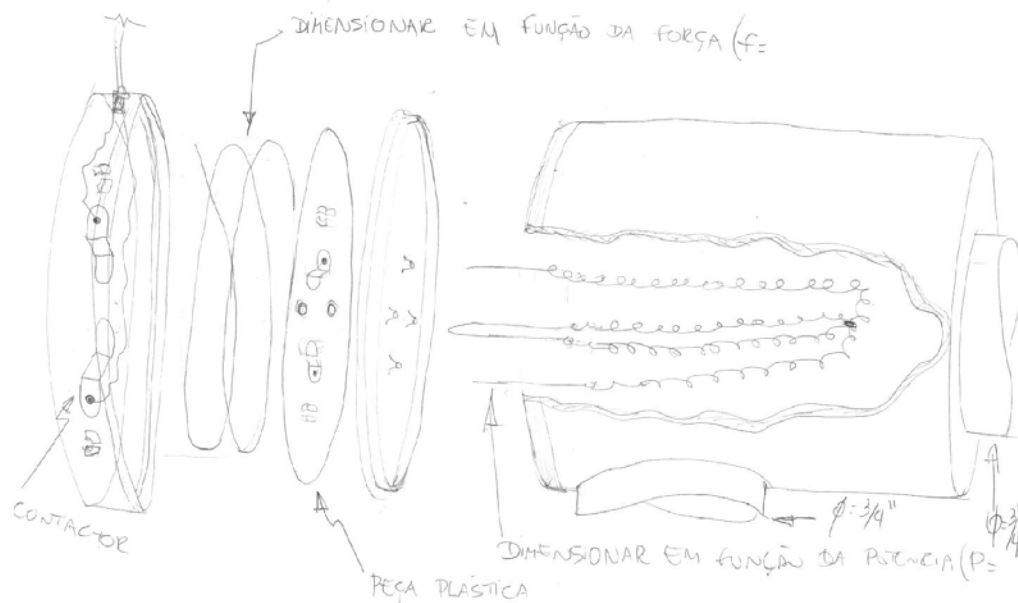


Figura 41. Vista explodida do sistema de aquecimento da água.

As várias peças do sistema estão ilustradas nas figuras abaixo, e cada uma é referenciada por objeto da classe **Part**.



Figura 42. Resistor

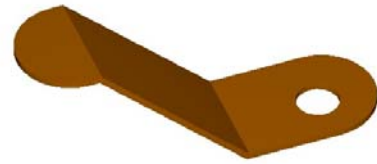


Figura 43. Contactor

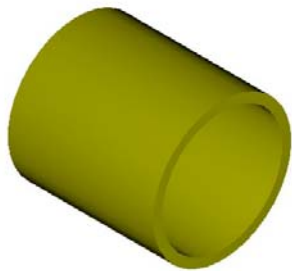


Figura 44. Isolante



Figura 45. Mola



Figura 46. Encosto anterior



Figura 47. Encosto posterior

A Figura 48 mostra a montagem do sistema de aquecimento, a qual é referenciada por um objeto da classe **Assembly**.

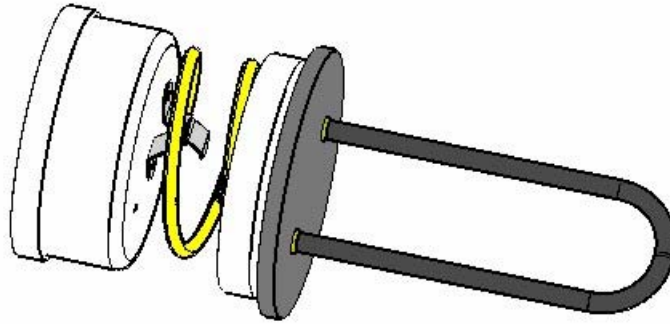


Figura 48. A montagem do sistema de aquecimento.

Capítulo 7

Conclusões e Considerações Finais

7.1. CONCLUSÕES E CONSIDERAÇÕES DA TESE

Algumas conclusões podem ser inferidas sobre a conveniência de adotar a modelagem de informações baseadas em *features* no processo de projeto de produto, e antever as suas vantagens quando aplicadas a sistemas computacionais de suporte à atividade de projeto em Engenharia.

- O modelo para representar o produto na atividade de projeto deve ser um modelo de *feature*.
- O modelo de informação usado para modelar o projeto deve ser um modelo de objeto.
- E o conceito de *feature* deve contemplar as informações geométricas e não-geométricas, e a semântica por trás das informações de projeto.

A principal contribuição foi a identificação, classificação e sistematização das principais *features* relacionadas a cada fase de projeto, implementadas no Capítulo 6. Tais *features* foram modeladas como classes

de objetos; e dessa forma, disponibilizadas como fonte de informação para o desenvolvimento de sistemas computacionais de suporte ao projeto, para a geração de projetos novos, bem como serem utilizadas na recuperação da intenção do projetista durante o reprojeto e reuso de informação do produto.

Como benefícios secundários ou relacionados, pode-se mencionar que:

- A identificação e evolução de *features* nas fases de projeto, incorporadas nas modelagens de *feature* e orientação a objeto, sinalizam para o projetista a possibilidade de integração de sistemas computacionais de suporte ao processo de projeto com outras ferramentas ou métodos utilizados no processo de projeto sistemático.

Como exemplo, cita-se (OGLIARI, 1999) que lista alguns desses métodos de sistemática de projeto, já normalmente usados na fase de projeto conceitual: questionário estruturado, pesquisa de mercado, casa da qualidade (QFD), síntese de funções, matriz morfológica e valoração das alternativas de solução. Esses métodos organizam as informações em tabelas de banco de dados relacionais.

A proposta é de definir as informações como classes ao invés de tabelas relacionais, e organizar a informação com base numa modelagem orientada a objetos. Isso permite armazenar as classes em banco de dados orientados a objetos, sem a necessidade de conversão dos dados (*de relacional para objetos*).

- As *features* de fases de projeto, com seus atributos, informam sobre o nível de abstração desses dados que estão sendo manipulados no estado atual do projeto, e permitem inferir sobre qual será a sua provável evolução numa fase posterior do processo.

Isso pode ser constatado no estudo de caso da seção 6.3.1, quando *features* instanciadas nas classes de necessidade, foram herdadas por requisitos de usuário, requisitos de projeto, e assim por diante; e os quais ajudam a compor a *feature* de especificações de projeto.

- É importante salientar que os três diferentes níveis de pesquisa citados no item 1.2, que comenta sobre o contexto e as motivações da tese, não são independentes. E por isso, a dificuldade de uma definição na análise e na modelagem das informações de projeto descritas nessa tese. É importante ficar ciente que as linhas de pesquisas e os trabalhos futuros nessa área não devem ser implementados desconsiderando a influência e os resultados de outras pesquisas, tais como a defendida nessa tese. Mesmo que as contribuições no estado da arte se limitem aos domínios particulares de cada um três níveis: básico (*modelagem de informações*), intermediário (*formas eficientes e de alto nível na representação da informação*), e final (*contexto de aplicação*).
- A otimização dos sistemas computacionais para suporte às atividades de projeto passa por uma solução também otimizada de formas eficientes de captura e representação das informações de projeto. Os progressos em pesquisas nessa área, levam à necessidade de integração entre diferentes sistemas computacionais para suporte ao processo de projeto. Isso é consenso universal e uma necessidade compartilhada por vários pesquisadores, desenvolvedores e usuários das tecnologias baseadas em computador.

7.2. DIFICULDADES DO PROCESSO DE MODELAGEM

Algumas dificuldades na conclusão do trabalho podem ser citadas:

- Pioneirismo e ineditismo em trabalhos dessa natureza de modelagem no âmbito da área de concentração de Projetos de Sistemas Mecânicos a qual esteve associada à tese. São poucos os trabalhos que tiveram como resultado e foco a modelagem de informações do problema em estudo. No âmbito do nosso programa de pós-graduação, tem-se a menção da dissertação de mestrado defendida em 2002 (LIMA, 2002), intitulada “*Modelagem de Informações para a Fase de Projeto Informacional de Produtos*”, o qual adotou uma

modelagem de informações baseada num modelo ER (Entidade/Relacionamento) gerado para descrever a fase informacional do processo de projeto.

- O amadorismo associado à incipiência em alguns domínios do conhecimento, e principalmente quanto ao pouco suporte e experiência de uso e aplicação em modelagem de informação num nível muito abstrato do processo de projeto, e utilização de ferramentas CASE (*Computer-Aided Software Engineering*) no suporte à atividade de modelagem.
- Formas de apresentação dos resultados da tese.

A dificuldade nesse tópico advém do fato de as classes implementadas serem de nível de abstração muito básico. Isso torna o grau de implementação computacional muito dificultado, pois as informações numa situação real de projeto são de mais alto nível de abstração, e exige-se a necessidade de ferramentas complementares ou auxiliares para realmente implementar algo prático e útil.

7.3. SUGESTÕES PARA TRABALHOS FUTUROS

Como contribuição relevante desse trabalho para outros futuros trabalhos extensíveis ou relacionados a esse, é importante que se tenha iniciado uma discussão, reflexão e alguns direcionamentos no sentido de apontar para a necessidade de mudança na forma de pensar e representar a estrutura de dados em que um sistema computacional de suporte ao processo de projeto de produto, tais como um sistema CAD, deve ser baseado e implementado.

Existe a grande necessidade de mais trabalhos nessa linha de pesquisa que se mostra ainda muito promissora e atual visto que ainda, como mostram os próprios resultados e conclusões desse trabalho, as definições das *features* de projeto para as fases do processo de projeto lidam com informações, conceitos e termos não universalmente e unicamente acordados e aceitos perante pesquisadores da área. E isso advém da própria

natureza da atividade de projeto, e das diversas sistemáticas empregadas e anunciadas na literatura.

Como sugestões de pesquisas que dêem continuidade e reforcem os resultados aqui obtidos tem-se:

Na linha de Sistemáticas de Projeto:

- Pode-se agora tomar as classes aqui definidas e estendê-las para domínios de aplicação específica, verificando quão genéricas ficaram as definições das classes.
- Identificação de padrões de relacionamentos existentes entre grupo de *features* de projeto, os quais possam ser modelados de forma genérica para diferentes sistemáticas de projeto.

Na linha de personalização de Sistemas CAD:

- Estudo das API's de sistemas CAD para a implementação e instanciação das definições de *features* na plataforma nativa do sistema e na interface de usuário.

Referências Bibliográficas

- ACHTEN, H.; LEEUWEN, J. V.. **A Feature-based Description for Design Processes: A Case Study**. 1998. URL: <http://ds.calibre.bwk.tue.nl/Research/publications>, OUTUBRO-2000.
- BACK, N.. **Metodologia de Projeto de Produtos Industriais**. Rio de Janeiro, Ed. Guanabara Dois, p. 389, 1983.
- BARBOSA, C. A. M.; DREUX, M.; FEIJÓ, B.. **An Architecture for the Design Entity**. Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. XXV, No. 1, p. 15-22, JAN-MAR/2003.
- BIDARRA, R.. **Validity Maintenance in Semantic Feature Modeling**. Delft, 1999, 147 p.. Tese – Technische Universiteit Delft.
- BIDARRA, R.; BRONSVOORT, W. F.. **Semantic feature modelling**. Computer-Aided Design, vol. 32, p. 201-225, Março-2000.
- BRONSVOORT, W. F.; JANSEN, F. W.. **Feature modeling and conversion – Key concepts to concurrent engineering**. Computers in Industry, vol. 21, p. 61-86, 1993.
- BRUNETTI, G.; GOLOB, B.. **A feature-based approach towards an integrated product model including conceptual design information**. Computer-Aided Design, vol. 32, p. 877-887, 2000.
- CASE, K; HOUNSELL, M. S.. **Feature modelling: a validation methodology and its evaluation**. Journal of Materials Processing Technology, vol. 107, p. 15-23, 2000.
- CHANG, T.-C.; WYSK, R. A.; WANG, H.-P.. **Computer-Aided Manufacturing**. Prentice Hall, 2a. edição, 672 p., Julho-1997.
- CUNHA, R. R. M.; DIAS, A.. **Estudo e Desenvolvimento de Metodologias na Troca de Dados em CAD/CAM**. Estudo Dirigido – Disciplina EMC 6601: Tópicos Especiais em Projeto de Sistemas Mecânicos, Florianópolis-SC, 139 p., Abril-2000a.
- _____. **Um Esboço de uma Sistemática para Suporte às Fases do Processo de Projeto Aplicando Sistemas CAD**. 9o. Congresso Chileno de Engenharia Mecânica - COCIM 2000; Valparaíso - Chile, 10-13/10/2000b.
- CUNNINGHAM, J. J.; DIXON, J. R.. **Designing with features: The origin of features**. Proceedings ASME Computers in Engineering Conference, San Francisco, p. 237-243, 1988.

- DAY, M. **Objects in CAD**. CADD & CADdesk – Web Resource, 2000, URL: www.edaltd.co.uk, JANEIRO-2000.
- DIXON, J. R.; DUFFEY, M. R.; IRANI, R. K.; MEUNIER, K. L.; ORELUP, M. F.. **A proposed taxonomy of mechanical design problems**. ASME Computers in Engineering Conference, ASME Press, 31-Julho / 4-Agosto, 1988.
- FERRÃO, P. D. M.; SEGATTO, F.. **Aparelho de Hidromassagem Portátil – Subprojeto do Sistema de Aquecimento**. Florianópolis, 2001, 24 p.. Projeto Final (Disciplina de Introdução ao Projeto e Manufatura Assistido por Computador) – Departamento de Engenharia Mecânica – Universidade Federal de Santa Catarina.
- FONSECA, A. J. H.. **Sistematização do Processo de Obtenção das Especificações de Projeto de Produtos Industriais e sua Implementação Computacional**. Florianópolis, 2000, 180 p.. Tese (Doutorado em Engenharia Mecânica) – Departamento de Engenharia Mecânica – Universidade Federal de Santa Catarina.
- HSU, W.; WOON, I. M. Y.. **Current research in the conceptual design of mechanical products**. Computer-Aided Design, vol. 30, No. 5, p. 377-389, 1998.
- JORNAL DA CIÊNCIA – JC E-MAIL. **O que é Informação e o que é Ciência da Informação?** Notícias de C&T – Serviço da SBPC, No. 1802, 4/Junho/2001.
- KUMAR, V.; BURNS, D.; DUTTA, D.; HOFFMANN, C.. **A framework for object modeling**. Computer-Aided Design, vol. 31, No. 9, p. 541-556, 1999.
- LEE, K.-S.; LEE, K.. **Framework of an evolutionary design system incorporating design information and history**. Computers in Industry, vol. 44, p. 205-227, 2001.
- LIMA, L. M. B.. **Modelagem de Informações para a Fase de Projeto Informacional de Produtos**. Florianópolis, 2002, 180 p.. Dissertação (Mestrado em Engenharia Mecânica) – Departamento de Engenharia Mecânica – Universidade Federal de Santa Catarina.
- MÄNNISTO, T; PELTOSEN, H.; MARTIO, A.; SULONEN, R.. **Modelling generic product structures in STEP**. Computer-Aided Design, vol. 30, No. 14, p. 1111-1118, 1998.
- MAZIERO, N. L.. **Um Sistema Computacional Inteligente de Suporte ao Projeto, Manufatura e Montagem de Peças Baseado em Features: Uma Abordagem com Sistemas Especialistas**. Florianópolis, 1998, 317 p.. Tese (Doutorado em Engenharia Mecânica) – Departamento de Engenharia Mecânica – Universidade Federal de Santa Catarina.
- MBANG, S.; STROEHLE, H.; HAASIS, S.. **Intelligent Feature Model for Flexible Manufacturing**. Proceedings of the 12th International

- Conference on Flexible Automation and Intelligent Manufacturing - FAIM, Dresden-Germany, p. 116-126, 15-17/Julho/2002.
- MCGINNIS, B. D.; ULLMAN, D. G.. **The Evolution of Commitments in the Design of a Component.** Journal of Mechanical Design, vol. 114, Março-1992. URL: www.engr.orst.edu/~ullman
- MCMAHON, C.; BROWNE, J.. **CAD CAM: Principles, Practice and Manufacturing Management.** Addison Wesley Longman, 2a. edição, 665 p., Agosto-1998.
- MONTENEGRO, F.; PACHECO, R.. **Orientação a Objetos em C++.** Editora Moderna, Rio de Janeiro - RJ, p. 394, 1994.
- MYUNG, S.; HAN, S.. **Knowledge-based parametric design of mechanical products based on configuration design method.** Expert Systems with Applications, vol. 21, No. 2, p. 99-107, 2001.
- NYQVIST, O; NIELSEN, J.. **Features as a Part of the Functional Interface Between Product and Resource: An Approach Based on Standards.** Proceedings of the International CIRP Design Seminar, Stockholm - Sweden, p. 181-186, 06-08/Junho/2001.
- OGLIARI, A.. **Sistematização da Concepção de Produtos Auxiliada por Computador com Aplicações no Domínio de Componentes de Plástico Injetados.** Florianópolis, 1999, 349 p.. Tese (Doutorado em Engenharia Mecânica) - Departamento de Engenharia Mecânica - Universidade Federal de Santa Catarina.
- OHSUGA, S.. **Toward intelligent CAD systems.** Computer-Aided Design, vol. 21, No. 5, p. 315-337, Junho-1989.
- PAHL, G.; BEITZ, W.. **Engineering design: a systematic approach.** Springer-Verlag, 2a. edição, Grã-Bretanha-Londres, 544 p., 1996.
- SALOMONS, O. W.; VAN HOUTEN, F. J. A. M.; KALS, H. J. J.. **Review of Research in Feature-Based Design.** Journal of Manufacturing Systems, vol. 12, No. 2, 27 p., 1993.
- SHAH, J. J.. **Assessment of features technology.** Computer-Aided Design, vol. 23, No. 5, p. 331-343, Junho-1991.
- SHAH, J. J.; MÄNTYLÄ, M.. **Parametric and Feature-based CAD/CAM: Concepts, Techniques, and Applications.** Nova York, John Wiley & Sons, 619 p., 1995.
- SHAH, J. J.; MATHEW, A.. **Experimental investigation of STEP Form-Feature Information Model.** Computer-Aided Design, vol. 23, No. 4, p. 282-296, Maio-1991.
- SHAH, J. J.; ROGERS, M. T.. **Expert form feature modelling shell.** Computer-Aided Design, vol. 20, No. 9, p. 515-524, 1988.

- SILVA J., A. C.. **Uma Abordagem para Criação e Compartilhamento de Dados de Peças Através da Integração CAD-RDBMS**. Florianópolis, Abril-2001, 119 p.. Dissertação (Mestrado em Engenharia Mecânica) – Departamento de Engenharia Mecânica – Universidade Federal de Santa Catarina.
- SRINIVASAN, V.. **A geometrical product specification language based on a classification of symmetry groups**. Computer-Aided Design, vol. 31, No. 11, p. 659-668, 1999.
- ULLMAN, D. G.; WOOD, S.; CRAIG, D.. **The Importance of Drawing in the Mechanical Design Process**. Computer & Graphics, vol. 14, No. 2, p. 263-274, 1990. URL: www.engr.orst.edu/~ullman, JULHO-2000.
- VAN VLIET, J. W.; VAN LUTTERVELT, C. A.; KALS, H. J. J.. **Three design tools to support the entire design process**. Proceedings of the International CIRP Design Seminar, p. 441-446, Estocolmo – Suécia, 06-08/Junho, 2001.
- WAKEFORD, L.; FAY, T.. **Choosing a CAD System with Automation in Mind**. Computer-Aided Engineering, Julho-1998.
- WARMAN, E. A.. **Object Oriented Programming and CAD**. Journal of Engineering Design, vol. 1, No. 1, p. 37-46, 1990.

Bibliografias

- ASHLEY, S.. **Manufacturing firms face the future.** Mechanical Engineering, p. 70-74, Junho-1997.
- AU, C. K.; YUEN, M. M. F.. **A semantic feature language for sculptured object modeling.** Computer-Aided Design, vol. 32, p. 63-74, 2000.
- BALIC, J.; BREZOCNIK, M.; CUS, F.. **Modeling of Mechanical Parts Compositions using Genetic Programming.** Trabalho aceito no FAIM – Flexible Automation And Intelligent Manufacturing, Maryland-USA, a ser realizado de 26-28/06/2000. URL: www.faim2000.isr.umd.edu, ABRIL-2000.
- BANKS, J.; GIBSON, R.. **Simulation Modeling – Some Programming Required.** IIE Solutions, p. 26-31, Fevereiro-1997.
- BECKERT, B.; KEMPFER, L.; SCHIMTZ, B.; DVORAK, P.. **1997 CAD/CAM – Planning Guide.** Um suplemento da Computer-Aided Engineering e Machine Design, p. 1-20, 1997.
- BERTOLINE, G. R.; WIEBE, E. N.; MILLER, C. L.; MOHLER, J. L.. **Technical Graphics Communication.** Irwin Graphics Series, 2a. edição, 1128 p., 1997.
- BOKULICH, F.. **CAD – Software Integration.** Automotive Engineering International, p. 52-64, Abril-1999.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.. **The unified modeling language user guide.** Addison-Wesley Inc., 482 p., 1999.
- CIMDATA, Inc. **Product Data Management: The Definition – An Introduction to Concepts, Benefits, and Terminology.** URL: www.CIMdata.com, Dezembro-1999.
- CUNHA, R. R. M.. **Padrão STEP para Transferência e Conversão de Dados.** Seminário apresentado na disciplina EMC 6634, Dezembro-1996.
- CUNHA, R. R. M.; DIAS, A.. **A Feature-Based Database Evolution Approach in Design Process.** Robotics and Computer Integrated Manufacturing, ISBN 0736-5845, vol. 18, No. 3-4, p. 275-281, 2002.
- _____. **Ambientes computacionais para integração de sistemas CAD/CAM.** Florianópolis-SC, 1999. Plano de Trabalho para Doutorado - Departamento de Engenharia Mecânica.

- _____. **Mechanical Design Data Approach Based on Objects and Features.** Proceedings of the International CIRP Design Seminar, Stockholm - Sweden, p. 295-300, 06-08/Junho/2001.
- _____. **Uma Revisão das Tecnologias de Integração de Dados em CAD/CAM.** Congresso Nacional de Engenharia Mecânica, Natal - RN, 07 a 11/08/2000.
- DIAS, A.. **An Iterative CAD/CAM System Approach for Robot's "Off-Line" Programming.** Relatório de Atividades do Pós-doutorado, ISE Dept., Virginia Polytechnic Institute - USA, 72 p., 1998.
- DIEHL, A.. **Transferring files from CAD to CAM.** Computer-Aided Engineering, p. 50-52, 1996.
- DVORAK, P.. **What moldmakers need from designers.** Machine Design, p. 103-105, Janeiro-1998. URL: www.penton.com/md
- EASTMAN, C. M.. **Out of STEP.** Computer-Aided Design, vol. 26, No. 5, p. 338-340, Julho-1994.
- EASTMAN, C. M.; FERESHETIAN, N.. **Information models for use in product design: a comparison.** Computer-Aided Design, vol. 26, No. 7, p. 551-572, Julho-1994.
- ERIKSSON, H.-E.; PENKER, M.. **UML Toolkit.** John Wiley Sons, Inc.; New York, 398 p., 1998.
- FARIN, G.. **Curves and Surfaces for CAGD - A Practical Guide.** Academic Press, New York, 4a. edição, 429 p., 1997.
- FOLEY, J. D.; DAM, A.; FEINER, S. K.; HUGHES, J. F.. **Computer Graphics - Principles and Practice.** Addison-Wesley Publishing Co., The Systems Programming Series, 2a. edição em C, 1996.
- FUH, J. Y. H.; CHANG, C.-H.; MELKANOFF, M. A.. **The development of an integrated and intelligent CAD/CAPP/CAFP environment using logic-based reasoning.** Computer-Aided Design, vol. 28, No. 3, p. 217-232, 1996.
- GOSLING, J.; MCGILTON, H.. **The Java Language Environment - A White Paper.** Maio-1996. URL: java.sun.com/doc/language_environment, AGOSTO-2000.
- GUERRERO, V.; ROZENFELD, H.. **Proposta de Classificação de Sistemas PDM.** XV Congresso Brasileiro de Engenharia Mecânica, Águas de Lindóia-SP, 22-26/11/1999.
- HARDING, J. A.; POPPLEWELL, K.; FUNG, R. Y. K.; OMAR, A. R.. **An intelligent information framework relating customer requirements and product characteristics.** Computers in Industry, vol. 44, No. 01, p. 51-65, Janeiro-2001.

- IBM, Inc. **An Overview of Object Oriented Programming.** URL: www.inf.ufsc.br/poo/smalltalk/ibm/tutorial/oop.html, JANEIRO-2000.
- JENSEN, C.; HINES, R.. **Interpreting Engineering Drawings.** Delmar Publishers, Inc., 400 p., 1989.
- JURISTO, N.; MORENO, A. M.. **How to Use Linguistic Instruments for Object-Oriented Analysis.** IEEE Software, p. 80-88, Maio/Junho-2000.
- KAFURA, D.. **Object-Oriented Software Design & Construction with JAVA.** Prentice Hall, New Jersey, 655 p., 2000.
- KERN, V. M.; BØHN, J. H.. **STEP Databases for Product Data Exchange.** First International Congress of Industrial Engineering, São Carlos-SP, Brasil, 4-7/Setembro, 1995.
- KIM, Won; BANERJEE, Jay; CHOU, Hong-Tai; GARZA, Jorge F.. **Object oriented database support for CAD.** Computer-Aided Design, vol. 22, No. 8, p. 469-479, Outubro-1990.
- KORTH, H. F.; SILBERSCHATZ, A.. **Sistema de Banco de Dados.** McGraw-Hill, Inc.; São Paulo, 581 p., 1989.
- KRAUSE, F.-L.; KIMURA, F.; KJELLBERG, T.; LU, S. C.-Y.. **Product Modelling.** Annals of the CIRP, p. 695-705, vol. 42/2/1993.
- KUMARA, S.; HAM, I.; AL-HAMANDO, M.; GOODNOW, K.. **Causal Reasoning and Data Abstraction in Component Design.** Annals of the CIRP, p. 145-148, vol. 38/1/1989.
- LEITE, L. L. P.. **Introdução aos Sistemas de Gerência de Banco de Dados.** Editora Edgard Blücher, São Paulo, 3a. reimpressão, 138 p., 1988.
- LIANG, J.; SHAH, J. J.; SOUZA, R. D.; URBAN, S. D.; AYYASWAMY, K.; HARTE, E.; BLUHM, T.. **Synthesis of consolidated data schema for engineering analysis from multiple STEP application protocols.** Computer-Aided Design, vol. 31, p. 429-447, 1999.
- LIU, S.; SOONG, G. E.. **Modeling Workflows with Reactive Objects.** Proceedings of FAIM, 10th International Conference, College Park, Maryland-USA, vol. 1, p. 623-630, Junho-2000.
- LOFFREDO, D.. **Efficient Database Implementation of EXPRESS Information Models.** New York, 1998. PhD Thesis, Rensselaer Polytechnic Institute. URL: www.steptools.com/~loffredo, AGOSTO-1999.
- MÄNTYLÄ, M.. **An Introduction to Solid Modeling.** Computer Science Press, 401 p., 1988.
- NARDELLI, E. S.. **Diversidade de materiais ajuda na concepção de projetos.** Revista CADesign, Market Press, Ano 5 - No. 54, 1999.

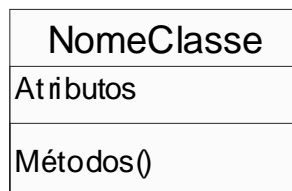
- OZAWA, M.; CUTKOSKY, M. R.; HOWLEY, B. J.. **Model Sharing among Agents in a Concurrent Product Development Team**. Submetido ao IFIP Working Group 5.2, Third Workshop KIC-3, Dezembro-1998.
- PIERRA, G.; AIT-AMEUR, Y.; BESNARD, F.; GIRARD, P.; POTIER, J-C.. **A general framework for parametric product model within STEP and parts library**. Artigo a ser apresentado no European Conference Product Data Technology, Londres. URL: www.nist.gov/sc4, FEVEREIRO-2000.
- REGLI, W. C.; CICIRELLO, V. A.. **Managing digital libraries for computer-aided design**. Computer-Aided Design, vol. 32, p. 119-132, 1999.
- RODRIGUEZ, L. C.; TSUZUKI, M. S. G.. **“FEATURES”**: Um Novo Paradigma em Sistemas CAD/CAM. Anais de resumos e CDROM do XVI Congresso Brasileiro de Engenharia Mecânica – COBEM, Uberlândia – MG, 10 p., 26-30/Novembro/2001.
- ROGERS, D. F.; ADAMS, J. A.. **Mathematical Elements for Computer Graphics**. McGraw-Hill, Inc; New York, 2a. edição, p. 611, 1990.
- ROY, U.; PANAYIL, D.. **Development of a Feature-based Rapid Design Environment**. Comput. Appl. Eng. Educ., John Wiley & Sons, vol. 5, p. 41-60, 1997.
- RUMBAUGH, J.; BLAHA, M.; PREMERLANI, W.; EDDY, F.; LORENSEN, W.. **Object-Oriented Modeling and Design**. Prentice Hall, Inc.; New Jersey, 500 p., 1991.
- SALOMONS, O. W.. **Computer Support in the Design of Mechanical Products – Constraint specification and satisfaction in feature based design for manufacturing**. Twente, 1995. PhD Thesis, Department of Mechanical Engineering, Universiteit Twente. URL: www.pt.wb.utwente.nl/staff/otto/thesis, SETEMBRO-1999.
- SC4 Home Page. **Welcome to ISO TC184/SC4 Developers of International Industrial Data Standards**. URL: www.nist.gov/sc4, FEVEREIRO-2000.
- SCHÜTZER, K.; GLOCKNER, C.; BATOCCHIO, A.. **Implementação e Testes de um Ambiente Integrado de Projeto Baseado em “Manufacturing Features”**. XV COBEM – Congresso Brasileiro de Engenharia Mecânica, Águas de Lindóia - SP, 22-26/Novembro, 1999.
- SILVA J., A. C.; CUNHA, R. R. M.. **Problemas e Princípios de Solução na Integração CAD/CAM**. Seminário apresentado na disciplina EMC 6327, Setembro-1999.
- SILVA J., A. C.; DIAS, A.. **Desenvolvimento de um Sistema para Troca de Dados em Projeto Mecânico**. Estudo Dirigido – Disciplina EMC 6601: Tópicos Especiais em Projeto de Sistemas Mecânicos, Florianópolis-SC, 98 p., Abril-2000.

- SMALL, C. H.. **Forging links in the CAE, CAD, CAM chain requires new methodologies.** Computer Design, p. 80-85, Setembro-1998.
- SPATIAL Technology, Inc.. **ACIS 3D Toolkit – SAT Format.** URL: www.spatial.com, 1999.
- STAIR, R. M.. **Princípios de Sistemas de Informação – Uma Abordagem Gerencial.** Thompson Publishing Inc., 2a. edição, 451 p., 1998.
- STEPTOOLS, Inc.. **What is STEP?** URL: www.steptools.com/library/about_step.html, FEVEREIRO-1999.
- ULLMAN, D. G.. **Toward the Ideal Mechanical Engineering Support System.** URL: www.engr.orst.edu/~ullman; www.wbh.com/cadsociety/wholepage_summit.html, JULHO-2000.
- ULLMAN, D. G.; DIETTERICH, T. G.; STAUFFER, L. A.. **A Model of the Mechanical Design Process Based on Empirical Data.** Academic Press Limited, AI EDAM, vol. 2 (1), p. 33-52, 1988. URL: www.engr.orst.edu/~ullman, JULHO-2000.
- ZEID, I.. **CAD/CAM Theory and Practice.** McGraw Hill, Inc.; New York, 1052 p., 1991.

APÊNDICE 1

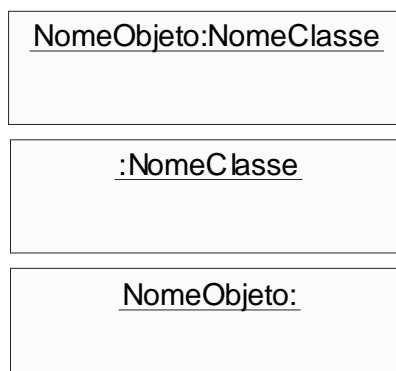
Notação e Simbologia UML

REPRESENTAÇÃO DE CLASSES:



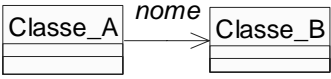
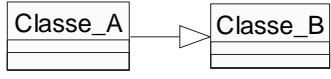
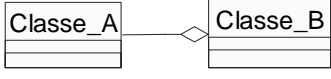
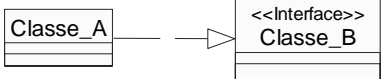
A representação consiste num retângulo com três áreas compartimentos distintos: o primeiro guarda o nome da classe; o segundo lista todos os atributos da classe; e o último lista todos os métodos.

REPRESENTAÇÃO DE OBJETOS:



Existem as três formas acima possíveis de representar um objeto em modelagem. Duas enfatizam o nome do objeto instanciado, enquanto a outra maneira que apenas enfatiza o nome da classe é usada nos casos onde quer se referir a objetos da classe em geral.

REPRESENTAÇÃO DOS RELACIONAMENTOS ENTRE CLASSES:

	Associação	Classe_A está associada a Classe_B
	Generalização / Especialização	Classe_A é do tipo Classe_B e herda as características da Classe_B
	Agregação	Classe_B possui a Classe_A
	Interface	Classe_B é uma interface para a Classe_A

APÊNDICE 2

Pesquisa sobre o Conceito de Feature

A pesquisa consistiu no envio por e-mail de um questionário contendo 3 (três) perguntas sobre *feature* a pesquisadores brasileiros no âmbito das áreas de projeto mecânico e fabricação. Essas perguntas são listadas abaixo, juntamente com as respostas dos pesquisadores que responderam, e retornaram o questionário também por e-mail.

O número de respondentes – limitado a 3 (três) apenas – não caracteriza uma amostragem representativa para uma análise e validação científica. Apesar disso, o resultado está citado com o objetivo de acrescentar mais algumas opiniões e compreensões sobre o conceito de *feature*, além daquelas obtidas nos artigos referenciados na secção 3.2.2 da tese.

PERGUNTAS		RESPONDENTES		
		1	2	3
1	No seu entendimento, o que é o conceito de Feature?	<p><i>“Features são “feições geométricas” características de processos industriais, desde a fase de design até as fases de planejamento de processo e manufatura. Features são uma forma de interfaceamento com um sistema gráfico modelador geométrico, de alto grau de abstração e grande facilidade que deve estar relacionada com o uso de seu significado para a integração com outras atividades da manufatura.”</i></p>	<p><i>“Feature é um elemento importante para o profissional que está tentando solucionar um determinado problema em sua área específica. Por exemplo, para um projetista, uma feature é um elemento que o auxilia a efetuar o projeto do produto (p. ex: um furo, um chanfro, etc.). Para o engenheiro de manufatura que trabalha com processos de usinagem, ele estaria interessado em identificar o material que ele teria que remover para chegar à peça especificada no desenho de projeto. Para o engenheiro do setor de qualidade, no caso do uso de uma máquina tridimensional (“CMM”), ele estaria interessado em verificar a conformidade ou não de superfícies da peça às exigências de projeto, e portanto um exemplo de feature poderia ser um cilindro, um cone ou uma superfície plana. Na minha opinião é um equívoco tentar considerar que somente uma definição de feature possa ser aplicada a diferentes áreas de uma empresa.”</i></p>	<p><i>“Feature é uma unidade de informação que contém algumas características básicas: é mapeável para uma forma (geométrica) genérica, tem um significado de engenharia, e é fortemente dependente da perspectiva do usuário, do tipo de produto em que se aplica, e da aplicação. Sua principal função é criar associatividade entre os elementos e permitir o reuso de informações pré-estabelecidas.”</i></p>

PERGUNTAS		RESPONDENTES		
		1	2	3
2	Em que palavra você resumiria o seu entendimento?	<i>“Features são “feições geométricas” que tem em si um significado de engenharia, e não somente geométrico.”</i>	<i>“Talvez a palavra “elemento”. Mas considero esta palavra um tanto genérica e, além disso, mesmo considerando uma atividade específica (p. ex. projeto do produto), devem ser considerados os “relacionamentos” deste “elemento” com os outros “elementos” no projeto.”</i>	<i>“associação”</i>
3	Por quê?	<i>“De um modo geral espera-se que as feições possam identificar ou registrar as “intenções dos projetistas” para que estas possam ser usadas em futuras tomadas de decisões sobre a manufatura, de forma automática preferencialmente, e também como recurso de documentação.”</i>	<i>“Não encontro outra palavra mais próxima.”</i>	<i>“Porque esta é a função principal de uma feature: servir como componente de ligação entre elementos distintos para enriquecer o modelo do produto.”</i>

