

**JOSÉ EDUARD ZINDEL DEBONI**

**UM MÉTODO PARA INTEGRAÇÃO DE SISTEMAS DE  
APOIO A ENGENHARIA ATRAVÉS DA REUTILIZAÇÃO  
DE SOFTWARE**

**Tese apresentada à Escola  
Politécnica da Universidade de São  
Paulo para obtenção do título de  
Doutor em Engenharia.**

**São Paulo  
1997**

**AGRADECIMENTOS**

Ao amigo e orientador Prof. Dr. José Sidnei Colombo Martini pela orientação segura e apoio constante.

Ao Centro Tecnológico da Marinha em São Paulo que motivou o desenvolvimento deste trabalho.

À minha mulher Teresa e meu filho Murilo pelo estímulo e compreensão.

Aos muitos colegas do Departamento de Informática pelas valiosas sugestões.

## SUMÁRIO

1. INTRODUÇÃO .....	1
1.1 Sistemas computacionais de apoio a projetos de engenharia .....	1
1.2 Definição do problema .....	3
1.3 Aplicabilidade e abrangência do trabalho .....	6
2. CARACTERÍSTICAS DA ATIVIDADE DE PROJETO EM ENGENHARIA .....	10
2.1 Introdução .....	10
2.1.1 Objetivos da automação de projetos .....	11
2.2 Evolução da automação de projetos de engenharia ...	12
2.2.1 Fase pioneira da automação de projetos .....	13
2.2.2 A fase do CAD/CAM .....	14
2.2.3 A fase do CIM .....	17
2.2.4 A fase da empresa virtual .....	19
2.2.5 A automação de projetos no Brasil .....	24
2.3 Modelagem do processo de projeto em engenharia ....	26
2.3.1 Técnicas para modelagem do processo .....	27
2.3.2 Modelos do processo de projeto de engenharia ...	29
2.3.3 Arquitetura geral de um sistema de projeto em engenharia .....	35
2.3.3.1 Camada de gerenciamento .....	37
2.3.3.2 Camada de desenvolvimento .....	39
2.3.3.3 Camada de infra-estrutura .....	39
2.4 Produtos de software para automação de projetos em engenharia .....	40
2.4.1 Integração de produtos de software .....	42
2.4.2 Sistemas operacionais para integração de software .....	43
2.4.3 Padrões para troca de informação entre produtos de software .....	46
2.4.4 Classificação dos produtos de software .....	49
2.5 Conclusões .....	50
3. DESCRIÇÃO DO MÉTODO PARA INTEGRAÇÃO DE SISTEMAS ....	52
3.1 Introdução .....	52

3.2	Justificativa da seleção do paradigma de objetos na modelagem de SCAPE .....	54
3.3	Técnicas de modelagem utilizadas pelo método .....	56
3.4	Fases do método de integração do SCAPE .....	58
3.5	Fase de Particionamento de Domínios .....	59
3.5.1	Identificação e Partic. de Domínios .....	61
3.5.2	Questionário de auxílio na análise de um SCAPE ..	62
3.5.3	Produtos da fase de partição dos domínios .....	63
3.6	Fase de análise dos domínios .....	63
3.6.1	Caracterização das classes e relac. ....	64
3.6.2	Produtos da fase de análise .....	69
3.7	Fase de projeto do SCAPE .....	70
3.7.1	Definição da plataforma de processamento .....	70
3.7.2	Detalhamento do diagrama de classes .....	72
3.7.3	Produto da fase de projeto de sistemas .....	73
3.8	Fase de implementação do SCAPE .....	74
3.8.1	Seleção das classes aqui implementadas .....	74
3.8.2	Tradução das classes em produtos de software ...	74
3.8.3	Produtos da fase de implementação .....	76
3.9	Resumo do método .....	77
4.	EXPERIMENTOS DE LABORATÓRIO E DE APLICAÇÃO DO MÉTODO .....	78
4.1	Descrição do laboratório de experimentos .....	78
4.2	Caso 1: Célula de trabalho para o engenheiro projetista .....	83
4.2.1	Particionamento dos domínios do caso 1 .....	84
4.2.2	Análise dos domínios do caso 1 .....	84
4.2.2.1	Diagrama de classes .....	88
4.2.3	Projeto do SCAPE para o caso 1 .....	89
4.2.4	Implementação do SCAPE no caso 1 .....	90
4.3	Caso 2: Sistema Computacional de Apoio ao Projeto de Navios .....	94
4.3.1	Particionamento dos domínios do caso 2 .....	94
4.3.2	Análise dos domínios do caso 2 .....	94
4.3.2.1	Análise do domínio do produto: navio .....	95
4.3.2.2	Análise do domínio do gerenciamento do estaleiro .....	95

4.3.2.3	Análise do domínio do desenvolvimento do navio .....	96
4.3.2.4	Análise do domínio da infra-estrutura .....	100
4.3.3	Projeto do SCAPE do caso 2 .....	104
4.3.4	Implementação do SCAPE no caso 2 .....	106
4.4	Caso 3: Sistema computacional de apoio ao projeto de propulsão nuclear .....	111
4.4.1	Particionamento dos domínios do caso 3 .....	111
4.4.2	Análise do questionário de apoio à análise no caso 3 .....	113
4.4.3	Análise dos domínios do caso 3 .....	114
4.4.3.1	Análise do domínio do Produto .....	115
4.4.3.2	Análise do domínio do gerenciamento .....	115
4.4.3.3	Análise do domínio técnico .....	117
4.4.3.4	Análise do domínio de apoio administrativo ....	118
4.4.3.5	Análise do domínio da implementação .....	120
4.4.4	Projeto de modificações no SCAPE do caso 3 ...	122
4.4.5	Modificações no SCAPE do CTMSP .....	128
5.	CONCLUSÕES E PERSPECTIVAS FUTURAS .....	130
5.1	Conclusões .....	130
5.2	Perspectivas Futuras .....	133
Anexo A. DESCRIÇÃO DO BANCO DE DADOS DE PRODUTOS DE SOFTWARE .....		
		136
1.	Introdução .....	136
2.	Classificação das Ferramentas de Software .....	136
a)	Função .....	136
b)	Plataforma de Processamento .....	137
c)	Área de Aplicação .....	138
d)	Padrões seguidos .....	140
3.	Construção do BDSOFT .....	140
a)	Modelo de objetos do BDSOFT .....	141
b)	Implementação do BDSOFT .....	141
Anexo B. INTRODUÇÃO À MODELAGEM ORIENTADA POR OBJETOS .....		
		143
1.	Conceitos Fundamentais .....	143

2. O método OMT.....	145
3. Notação usada nos diagrama.....	146
a) Diagrama de Domínios.....	147
b) O Diagrama de classes.....	149
c) O diagrama de eventos.....	152
d) O diagrama de estados.....	154
e) O diagrama de fluxo de dados.....	154
4. Conclusão.....	156
Anexo C. QUESTIONÁRIO DE APOIO À ANÁLISE DE UM SCAPE..	158
1. Introdução.....	158
2. Construção do Questionário.....	158
3. Instruções para se responder o questionário.....	159
4. Análise das respostas.....	173
5. Programa ANALISE.....	175
Anexo D. INSTRUÇÕES PARA USO DO PROGRAMA BSCAPE.....	177
1. Introdução.....	177
2. Instalação do programa BSCAPE.....	177
3. Seqüência de passos para uso do BSCAPE.....	177
a) Criando o Banco de Produtos.....	177
b) Importando o diagrama de classes.....	178
c) Selecionando o sistema a ser implementado.....	179
d) Implementando uma função do diagrama.....	179
e) Emitindo os relatórios da implementação.....	181
4. Conclusão.....	181
Referências Bibliográficas.....	182

## LISTA DE FIGURAS

Figura 1 - Relação entre o processo de solução de um problema de engenharia e o SCAPE.....	2
Figura 2 - Organização geral do estudo.....	6
Figura 3- Modelo de camadas da organização de uma equipe virtual.....	31
Figura 4 Interação do modelo com as ferramentas de projeto.....	33
Figura 5 - Estrutura de integração de projeto.....	34
Figura 6 - Modelo de organização do ambiente de projeto.....	35
Figura 7 - Integração de ferramentas pela troca de arquivos.....	46
Figura 8 - Diagrama de domínios de um SCAPE.....	60
Figura 9 - Principais classes de um SCAPE.....	65
Figura 10 - Diagrama de classes típico do domínio do produto.....	66
Figura 11 - Diagrama de classes típico do domínio de gerenciamento.....	67
Figura 12 - Exemplo do diagrama de transição de estado de uma atividade.....	67
Figura 13 - Diagrama de classes típico do domínio de desenvolvimento.....	68
Figura 14 - Exemplo do diagrama de eventos de um processo.....	68
Figura 15 - Exemplo de classe típica do domínio de infra-estrutura.....	69
Figura 16 - Exemplo de operação nos diagramas para facilitar a tradução.....	73
Figura 17 - Exemplo de detalhamento de classe.....	73
Figura 18 - Exemplo de instanciação de produtos de software.....	75
Figura 19 - Relacionamento entre classes e produtos...	75
Figura 20- Esquema do Apoio Computacional do Laboratório de Experimentos.....	79
Figura 21 - Exemplo da interface do programa WITH CLASS 3.0.....	80
Figura 22 - Distribuição dos produtos cadastrados entre as funções.....	81
Figura 23 - Amostra da interface do programa ESCAPE...	82

Figura 24 - Diagrama de domínios da célula de trabalho .....	84
Figura 25 - Diagrama de eventos da solicitação de um Estudo .....	85
Figura 26 - Diagrama de eventos da preparação de uma proposta .....	86
Figura 27 - Diagrama de eventos da edição de um relatório técnico .....	87
Figura 28 - Diagrama dos eventos no domínio da infra-estrutura .....	87
Figura 29 - Diagrama de classes resultante da análise de domínios .....	88
Figura 30 - Diagrama de classes resultante do projeto.	90
Figura 31 - Diagrama de Classes final do sistema .....	93
Figura 32 - Diagrama de classes do domínio do produto.	95
Figura 33 - Diagrama de classes do domínio do gerenciamento .....	96
Figura 34 - Diagrama de eventos do processo de gerenciamento .....	96
Figura 35 - Diagrama de classes do domínio do desenvolvimento .....	97
Figura 36 - Diagrama dos eventos de projeto do casco ..	98
Figura 37 - Diagrama de eventos do projeto mecânico e elétrico .....	99
Figura 38 - Diagrama de eventos do projeto de tubulação .....	99
Figura 39 - Diagrama de eventos da análise da maquete .....	100
Figura 40 - Diagrama de classes do domínio da infra-estrutura .....	101
Figura 41 - Processo de gerenciamento dos documentos de projeto .....	101
Figura 42 - Processo de gerenciamento das listas de material .....	102
Figura 43 - Diagrama de classes resultante da análise de domínios .....	103
Figura 44 - Diagrama de classes resultante do projeto .....	107
Figura 45 - Diagrama de classes final do SCAPE no caso 2. ....	110
Figura 46 - Diagrama de domínios .....	112

Figura 47 - Quadro de respostas do questionário de análise .....	113
Figura 48 - Diagrama de classes do domínio do produto .	115
Figura 49 - Diagrama de classes do domínio do gerenciamento .....	116
Figura 50 - Diagrama de eventos do processo de gerenciamento .....	117
Figura 51 - Diagrama de classes do domínio do desenvolvimento .....	118
Figura 52 - Diagrama de classes do apoio administrativo .....	119
Figura 53 - Processo de gerenciamento dos documentos de projeto .....	120
Figura 54 - Diagrama de classes resultante da análise de domínios .....	123
Figura 55 - Diagrama de classes do domínio de gerenciamento na fase de projeto .....	123
Figura 56 - Diagrama de classe do domínio de apoio administrativo na fase de projeto .....	126
Figura 57 - Diagrama de classe do domínio técnico na fase de projeto .....	127
Figura 58 - Classes a serem implementadas no caso 3 ...	129
Figura A-1- Modelo de objetos do Banco de Produtos de Software .....	141
Figura A-2 - Interface de cadastro de empresas e produtos. ....	142
Figura B-1 - Evolução do sistema no método OMT .....	146
Figura B-2 - Exemplo de um diagrama de domínio (SHLAER; MELLOR, 1993) .....	148
Figura B-3 - Representação gráfica de uma classe e um objeto .....	149
Figura B-4 - Representação de uma classe .....	150
Figura B-5 - Representação da multiplicidade em uma associação. ....	150
Figura B-6 - Exemplo de atributo de ligação .....	151
Figura B-7 - Exemplo de estrutura de herança .....	151
Figura B-8 - Exemplo de agregação .....	152
Figura B-9 - Exemplo da notação de padrões .....	152
Figura B-10 - Diagrama de eventos para uma chamada telefônica .....	153
Figura B-11 - Exemplo de diagrama de estados como um laço contínuo. ....	154

Figura B-12 - Notação de Yourdon para os Diagramas de Fluxo de Dados .....	155
Figura B-13 - DFD para uma movimentação em ContraCorrente .....	156
Figura C-1 - Interface do programa ANALISE .....	176
Figura D-1 - Cadastrando produtos e empresas no Banco de Produtos .....	178
Figura D-2 - Opções do menu Sistemas no Programa BSCAPE .....	178
Figura D-3 - Interface para seleção do sistema implementado .....	179
Figura D-4 - Interface para seleção do sistema implementado .....	180
Figura D-5 - Seqüência de operações para implementar uma função .....	181

### LISTA DE ABREVIATURAS E SIGLAS

<b>Sigla</b>	<b>Significado</b>
APPLE	Empresa fabricante de microcomputadores
AUTODESK	Empresa desenvolvedora de software
CAD	Computer-Aided Design
CALS	Computer-Aided Acquisition and Logistic Suport
CAM	Computer-Aided Manufacturing
CASE	Computer-Aided Software Engineering
CERC	Concurrent Engineering Research Centre
CIM	Computer-Integrated Manufacturing
CIM-OSA	Computer-Integrated Manufacturing, Open Systems Association
COM	Common Object Model
COPESP	Coordenadoria para Projetos Especiais, órgão do Ministério da Marinha do Brasil
CTMSP	Centro Tecnológico da Marinha em São Paulo
DARPA	Defende Advanced Research Projects Agency
DICE	DARPA's Initiative on Concurrent Engineering
DXF	DATA EXCHANGE FORMAT, formato de arquivos gráficos
EMBRAER	Empresa Brasileira da Aeronáutica
ESPIRIT	Projeto tecnológico de consorcio de países europeus
EXPRESS	Linguagem de definição de dados associada ao padrão STEP
IBM	Empresa fabricante de computadores
ICAM	Integrated Computer-Aided Manufacturing, projeto da Força Aérea dos EUA
IDA	Institute for Defende Analisis
IDEF <sub>x</sub>	Integrated Definition, notação gráfica para análise desenvolvida pela Força Aérea dos EUA no projeto ICAM
IGES	INITIAL GRAPHICS EXCHANGE SPECIFICATION
INTEL	Empresa fabricante de microprocessadores
INTERNET	Rede mundial de computadores
ISO	International Standard Organization
NITS	National Institute os Standards and Technology
NOVELL	Empresa desenvolvedora de software de rede
OLE	Object Linking and Embedding
OPENDOC	Padrão para troca de

PDES	Product Data Exchange using STEP
SCAPE	Sistema Computacional de Apoio a Projetos de Engenharia
SGML	Standard Generalized Markup Language
SOM	System Object Model
STEP	STANDART FOR EXCHANGE OF PRODUCT DATA
WINDOWS	Sistema operacional para microcomputadores
WORDPERFECT	Empresa desenvolvedora de software

**RESUMO**

A automação de projetos é a área da automação que enfoca o desenvolvimento de sistemas computacionais de apoio a projetos de engenharia. Estes sistemas se caracterizam por serem formados pela integração de diversos produtos de software, disponibilizados pelo mercado de informática para este fim. Este trabalho trata da construção destes sistemas, e apresenta um método para a solução desta classe de problemas da engenharia de software.

Estuda-se a evolução histórica da automação de projetos para se chegar a uma arquitetura geral de um sistema automatizado de projetos de engenharia. Com base nesta arquitetura o método inova tanto na aplicação de técnicas de orientação por objetos para modelar o problema, como na classificação e reutilização de produtos de software para construir a solução.

Para viabilizar a utilização do método, foram desenvolvidos um banco de produtos de software e um programa de computador que apoia a fase de implementação do sistema, com base nos dados deste banco. Os resultados de sucesso de três casos de estudo comprovam a aplicabilidade do método.

**RESUMO**

A automação de projetos é a área da automação que enfoca o desenvolvimento de sistemas computacionais de apoio a projetos de engenharia. Estes sistemas se caracterizam por serem formados pela integração de diversos produtos de software, disponibilizados pelo mercado de informática para este fim. Este trabalho trata da construção destes sistemas, e apresenta um método para a solução desta classe de problemas da engenharia de software.

Estuda-se a evolução histórica da automação de projetos para se chegar a uma arquitetura geral de um sistema automatizado de projetos de engenharia. Com base nesta arquitetura o método inova tanto na aplicação de técnicas de orientação por objetos para modelar o problema, como na classificação e reutilização de produtos de software para construir a solução.

Para viabilizar a utilização do método, foram desenvolvidos um banco de produtos de software e um programa de computador que apoia a fase de implementação do sistema, com base nos dados deste banco. Os resultados de sucesso de três casos de estudo comprovam a aplicabilidade do método.

**"Abstract"**

Design automation is the part of automation that focus the development of computer-aided systems for engineering design. These systems are characterized of beeing formed by the union of many software packages, that are put available by the software market. This work addresses the problem of creating these systems, and it presents a method that solves this class of software engineering problems.

Studing the historic evolution of design automation, one can find a generic architecture for na automated engineering design system. Based on this architecture the method inovates either in the application of object-oriented technique to model the problem, either in the classification and reuse of software packages to build the solution.

In order to be capable of using the method, a database of software packages was organized and a computer program to support the system implementation phase, based on the information on the database, was developed. The successful results of three case-studies guarantee the applicability of the method.

## 1. INTRODUÇÃO

A tentativa de obter melhores soluções para os problemas de engenharia, levam à utilização, em larga escala, do computador como ferramenta de trabalho. A multidisciplinaridade e a crescente complexidade destes problemas vem exigindo uma quantidade cada vez maior de programas para resolvê-los. Este trabalho apresenta um método para o engenheiro de software realizar a integração dos programas, e dar o apoio computacional aos projetos de engenharia.

### 1.1 Sistemas computacionais de apoio a projetos de engenharia

A solução de um problema de síntese em engenharia pode ser dividida, simplifcadamente, em duas etapas: projeto e produção. A etapa de projeto parte dos requisitos definidos pelo problema para criar os documentos de projeto, que são utilizados na etapa de produção, para construir a solução desejada. Sistemas computacionais podem apoiar as tarefas desempenhadas pelos engenheiros durante as duas etapas.

Este trabalho aborda, exclusivamente, a automação das atividades que ocorrem na etapa de projeto. A chamada automação de projetos não visa substituir o engenheiro pelo computador, mas sim valorizar o seu trabalho. Objetiva-se,

principalmente, aumentar a capacidade humana de trabalho, através dos sistemas computacionais de apoio a projetos de engenharia, que aqui serão chamados, abreviadamente, por SCAPE.

A figura 1 mostra a relação do processo de solução de um problema de engenharia, com o sistema computacional de apoio à etapa de projeto. Observa-se o papel de integrador do engenheiro de software, selecionando os programas de computador do mercado, e construindo o SCAPE para os projetistas.

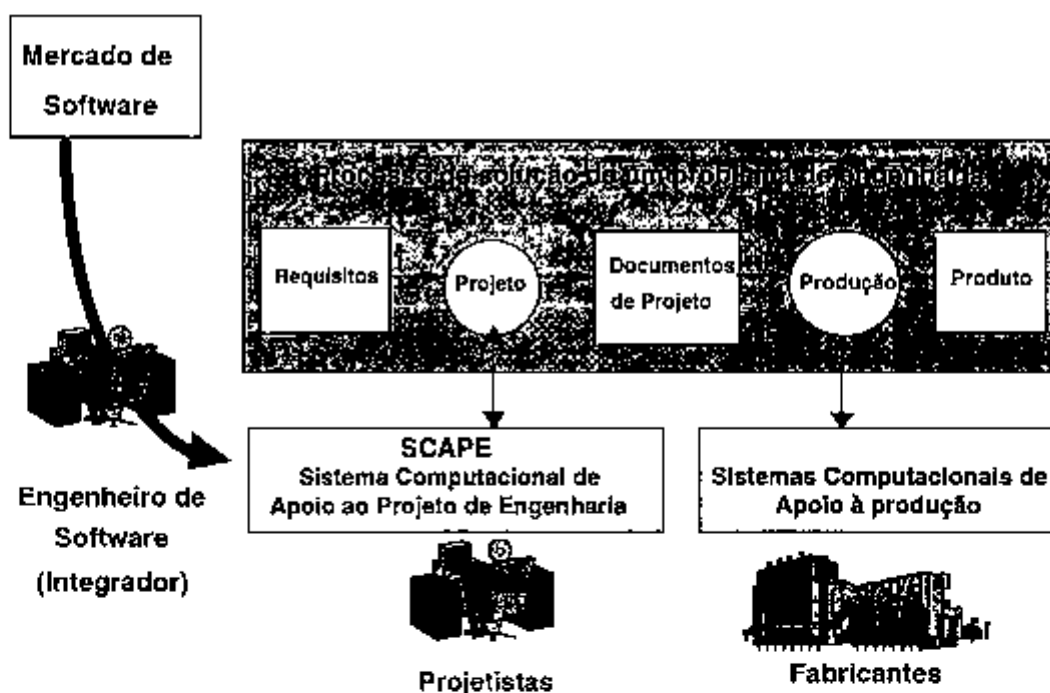


Figura 1 - Relação entre o processo de solução de um problema de engenharia e o SCAPE.

A demanda por SCAPE fez surgir um segmento específico na indústria de software. Para dar apoio às diversas especialidades da engenharia, uma grande variedade de programas de computador foram criados e estão sendo

aprimorados continuamente. Um sistema de automação de projetos utiliza, em geral, uma variedade destes programas. A decisão de quais programas utilizar e como integrá-los ao ambiente de trabalho é muito importante para o sucesso da automação.

## **1.2 Definição do problema**

As particularidades de cada SCAPE impedem que os métodos de engenharia de software existentes sejam aplicados diretamente. Os métodos disponíveis tratam do desenvolvimento de software imaginando que a implementação ocorrerá através de uma linguagem de programação (PRESSMAN, 1995). A utilização de produtos de software prontos, como recurso para a produção de um sistema, ainda não é muito explorada. Não existem métodos que abordem a reutilização de produtos de software como estratégia para atender as necessidades dos sistemas de um modo geral, tão pouco dos SCAPE, em particular.

As primeiras tentativas para sistematizar a construção de sistemas de apoio à engenharia, substituíam cada etapa do processo manual por um correspondente automatizado. Não explorou-se as características de integração e inter-relacionamento entre atividades decorrentes da automação (ARAKAKI, 1991).

Os diversos resultados das pesquisas realizadas sobre a implantação de SCAPE são importantes para o levantamento

dos requisitos de um sistema de automação de projetos. Estes estudos se concentram mais nos aspectos gerenciais e organizacionais do uso desta tecnologia, do que nos problemas computacionais envolvidos. (REDDY et al., 1993; ROZENFELD et al., 1992; CARTER, BAKER; 1991).

Trabalhos mais recentes definem as etapas iniciais do desenvolvimento de sistemas de automação de um modo geral, destacando o papel fundamental dos sistemas abertos na integração dos seus componentes. Não consideram, porém, com deverá ser sua implementação (TANOMARU; 1994).

A automação de projetos pode ser também considerada como uma especialização da automação de escritórios. Em seu estudo DUNCAN (1995) define o escritório como um "ambiente de trabalho, onde o insumo é a informação e o veículo para representar a informação é o documento". Sugere a aplicação de técnicas de engenharia de software baseadas em objetos para a análise do domínio do problema. Explora, no entanto, apenas a modelagem da interação que ocorre entre os usuários e o sistema de automação, e não como deve ser construído o sistema computacional de apoio.

Este trabalho apresenta um método de engenharia de software específico para a integração de sistemas computacionais de apoio à engenharia, baseado na reutilização de produtos de software. O método cobre o ciclo completo de vida de um SCAPE, da análise e concepção até a implementação da solução. A aplicação do método, por parte dos engenheiros

de software, possibilita construir, criteriosamente, um sistema para automação de projetos de engenharia.

O método apresentado é considerado um método de integração pois, ao contrário dos demais, o sistema é criado a partir da integração de programas de computador prontos, e não da compilação de um programa-fonte. Esta característica é inédita entre os métodos de engenharia de software.

O trabalho parte da identificação de uma arquitetura geral para os SCAPE. Esta arquitetura fornece por um lado os requisitos para a análise do problema de automação, e por outro permite a criação de uma biblioteca de produtos de software para serem reutilizados. Utiliza-se a tecnologia de objetos tanto para a criação dos modelos de análise de um problema real como para a concepção da biblioteca de produtos de software. A utilização do mesmo paradigma nestas duas fases, permite que a implementação de um SCAPE se dê pela tradução dos modelos de análise em produtos de software.

A figura 2 mostra a organização geral do trabalho, e faz um paralelo com a divisão temática dos capítulos.

O capítulo 2 apresenta as características que diferenciam um SCAPE de outros sistemas computacionais, estuda a sua evolução histórica, e propõe uma arquitetura geral para este tipo de sistema. O capítulo 3 apresenta o método propriamente dito; nele estão descritos, detalhadamente, os

passos necessários para a análise, projeto e implementação de um SCAPE. O capítulo 4 apresenta exemplos de utilização do método. As considerações finais e conclusões estão no capítulo 5. Os anexos tratam da especificação do banco de dados de produtos de software (Anexo A), de um questionário de apoio à análise de um SCAPE (Anexo B), de um resumo da notação utilizada pelos modelos orientados por objetos (Anexo C) e do uso do programa de computador desenvolvido para dar apoio método (Anexo D).

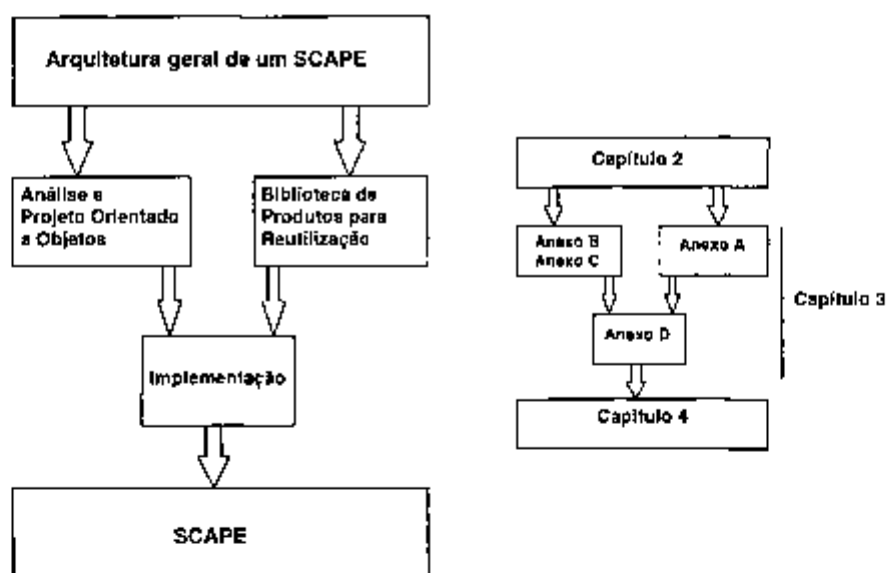


Figura 2 - Organização geral do estudo

### 1.3 Aplicabilidade e abrangência do trabalho

Pressionadas por um mercado mais competitivo, as empresas de engenharia buscam, continuamente, formas de reduzir seus custos e melhor atender às necessidades dos seus clientes. Este trabalho aplica-se a qualquer empresa de engenharia que procura reduzir seus custos e aumentar a qualidade dos

produtos, pela integração de uma SCAPE ao seu processo de desenvolvimento. Manter-se à frente no mercado, e responder às pressões de um ambiente em transformação, exige estar em constante melhoria, integrando aos seus métodos de trabalho a mais moderna tecnologia disponível. Diversos autores analisaram a importância da automação de projetos nas empresas.

CARTER; BAKER (1991) observam que nas empresas de engenharia todo estímulo externo deve ser dominado por alterações internas no ambiente de trabalho. Os sistemas de informação são os principais agentes destas alterações, respondendo às pressões, e viabilizando as mudanças. Concordam com esta afirmação HAMMER; CHAMPY (1993), no seu conhecido livro sobre "Re-engenharia". Consideram que são três as principais forças modificadoras do ambiente empresarial: os clientes, a concorrência e as mudanças tecnológicas. Concluem que caso a empresa não se mantenha alerta para estas novas condicionantes, corre o risco de ser surpreendida pela concorrência, sem tempo para reagir.

SCHWAB et al. (1993) observam que as exigências dos consumidores provocam uma redução na vida útil dos produtos, aproximando-a do seu próprio ciclo de desenvolvimento. Como a vida útil de um produto depende, exclusivamente, de forças do mercado, deve-se procurar reduzir o ciclo de desenvolvimento, com investimentos

estratégicos em automação, para manter a vitalidade da indústria.

Uma tecnologia em transformação, como é a tecnologia computacional, exige uma permanente atenção ao mercado de informática e um constante aperfeiçoamento técnico. Tais preocupações desviam o engenheiro de projetos da sua especialidade na tentativa de construir os sistemas computacionais de apoio (WEISBERG, 1995). Cabe ao engenheiro de software preparar o ambiente computacional para ser utilizado pelas outras especialidades, e garantir que os requisitos de qualidade de projeto e produtividade esperados estejam sendo satisfeitos.

A motivação para o desenvolvimento do método aqui apresentado surgiu durante a experiência profissional do autor junto ao Centro Tecnológico da Marinha em São Paulo (CTMSP), na tentativa de se criar um ambiente integrado de projetos nucleares (COPESP, 1992 e 1995). Nestes trabalhos foram aplicados modelos de engenharia de software na representação das relações existentes em um ambiente de projetos de engenharia.

De um modo geral, qualquer projeto de engenharia pode ser atendido por este método. No entanto, os benefícios são proporcionais à complexidade do projeto. Projetos multidisciplinares, que envolvem um grande número de profissionais, obtém maiores benefícios de um gerenciamento adequado dos seus recursos computacionais. Este é o caso de

projetos de grandes instalações industriais, projetos de embarcações e projetos nucleares. No entanto não está restrita a aplicação do método em outras áreas, ou em projetos de menor porte.

O método aqui proposto pode ser generalizado para qualquer sistema de software que se adapte à técnica de construção por integração de componentes previamente construídos. Neste caso deve-se identificar a arquitetura geral da classe de problemas desejada para orientar a análise e construir o banco de dados de componentes. Esta generalização foge do escopo deste trabalho.

A especialização em automação de projetos e o uso deste método, permitem que o engenheiro de software mantenha as empresas de engenharia em permanente sintonia com a tecnologia computacional disponível. A constância da evolução tecnológica e o potencial ainda não explorado da computação em projetos de engenharia, garantem ainda muitas mudanças por ocorrer.

## **2. CARACTERÍSTICAS DA ATIVIDADE DE PROJETO EM ENGENHARIA**

Este capítulo descreve as características da atividade de projeto em engenharia que são importantes para sua automação. Como resultado de uma revisão bibliográfica, caracterizam-se quatro fases na evolução da automação de projetos. Da análise das ferramentas computacionais e métodos utilizados em cada fase resulta um modelo geral da arquitetura de um sistema automatizado de projeto.

### **2.1 Introdução**

BAZZO; PEREIRA (1993) definem o projeto de engenharia como o "conjunto de atividades que precede a execução de um produto, sistema, processo ou serviço". Pode-se assim afirmar, que projetar é estabelecer, e documentar, o que deve ser executado para se obter um produto, sistema, processo ou serviço desejado.

Um projeto se inicia com a identificação de uma necessidade, e se encerra com a comunicação do que é preciso executar para se atender àquela necessidade. A realização da necessidade ocorre, após o projeto, na etapa de produção. A solução de um problema de engenharia fica então dividida nas etapas de projeto e produção. Este trabalho trata da fase de projeto e da sua automação.

Automação de projetos é a implantação apropriada de recursos computacionais nas atividades de projeto.

### **2.1.1 Objetivos da automação de projetos**

O engenheiro é na sua essência um projetista. Ao se automatizar as atividades de projeto, busca-se dotar o engenheiro de meios para poder melhor desempenhar a sua vocação. O uso de sistemas computacionais nas empresas de engenharia contribui para conquistar o mercado e obter maiores lucros, aumentando a eficiência dos projetistas, e melhorando a qualidade do seu trabalho.

A automação de projetos pode aumentar a produtividade dos projetistas, reduzindo o tempo de projeto, quer pelo aumento da velocidade de trabalho, quer pela diminuição da ocorrência de erros de projeto, e conseqüente redução no trabalho gasto para corrigi-los. O acréscimo de velocidade é conseguido simplificando-se etapas no processo, e utilizando-se a tecnologia computacional para acelerar a produção dos documentos de projeto. A redução de erros é resultado de uma maior integração dos projetistas com o projeto, obtida pelo uso de meios eletrônicos de comunicação e documentação. Pode-se reproduzir no computador o comportamento de um produto projetado, antevendo problemas, e testando possíveis soluções. Os efeitos positivos combinados da automação de projetos levam

a produtos de melhor qualidade, e com melhor competitividade no mercado.

Deve-se observar que não é objetivo da automação de projetos eliminar o projetista do processo de desenvolvimento, mas criar um ambiente no qual as suas atividades possam ser desempenhadas com maior facilidade, precisão e rapidez.

## **2.2 Evolução da automação de projetos de engenharia**

O desenvolvimento da computação gráfica permitiu a aplicação mais intensa do computador na engenharia, e a automação das atividades de projeto. Pode-se identificar quatro fases distintas na evolução da automação de projetos:

- Fase pioneira da automação de projetos,
- Fase do CAD/CAM,
- Fase do CIM, e
- Fase da empresa virtual.

Cada fase destaca um diferente aspecto do problema de automação, que naquele momento era considerado o mais importante, orientando o desenvolvimento de ferramentas de hardware e software para atendê-lo. Ao serem vencidas as dificuldades de uma fase, novos desafios se apresentam.

aumentando o grau de complexidade dos sistemas de apoio a projetos de engenharia.

### **2.2.1 Fase pioneira da automação de projetos**

A engenharia sempre foi uma grande usuária de computadores, sendo impossível precisar o primeiro caso onde o computador foi aplicado em um projeto de engenharia. Entretanto, somente após o surgimento da computação gráfica, eles se tornam ferramentas de apoio a todas as atividades dos engenheiros.

Nos anos 50, enquanto ainda se desenvolvia o embrião da computação gráfica, ela foi utilizada para apoiar a interface homem-máquina de sistemas militares norte-americanos de controle do espaço aéreo, no projeto SAGE (Semi-Automatic Ground Environment) (TEICHOLZ; 1985).

Durante a década de 60, o computador foi aplicado em projetos pelas grandes corporações da indústria aeronáutica e automobilística. Com apoio da indústria de computadores da época, surgiram os primeiros sistemas comerciais para automação de projetos, voltados para atender as necessidades de manufatura. Estes sistemas permitiam o desenho do perfil de uma peça mecânica simples na tela do computador, que comandava sua fabricação nas primeiras máquinas controladas numericamente.

Os sistemas comerciais para desenho automatizado se desenvolveram mais intensamente durante os anos 70,

baseados em computadores de propósito geral, aos quais se agregavam periféricos especializados como o terminal gráfico interativo, o traçador gráfico (plotter), a caneta ótica e a mesa digitalizadora. TEICHOLZ (1985) apresenta um resumo bastante completo das primeiras fases da automação de projetos.

### **2.2.2 A fase do CAD/CAM**

A fase seguinte na evolução da automação de projeto, iniciou-se com aparecimento de sistemas comerciais de CAD, a partir dos anos 70. Procurou-se adaptar o computador em cada etapa do processo de desenvolvimento de um produto. Os sistemas CAD/CAM, como eram conhecidos, formaram-se pela união de ferramentas de CAD (Computer-Aided Design), usadas na concepção e no desenho de detalhamento, com os programas de CAM (Computer-Aided Manufacturing) que ligavam o projeto com as máquinas de comando numérico da produção. Formaram-se as chamadas ilhas de automação. Algumas atividades recebiam recursos de automação, e se sobressaíam em meio a um mar de processos não automatizados. Se medidos isoladamente os índices de produtividade conseguidos eram altos, mas nem sempre eles se refletiam no processo de desenvolvimento do produto como um todo.

O termo CAD/CAM, empregado até hoje, identificou tanto as ferramentas utilizadas para apoio à engenharia, como o próprio processo de projeto automatizado. Este termo é

utilizado com freqüência como um sinônimo da própria automação de projetos.

O uso do computador em projetos de engenharia, desde o seu início, teve um crescimento exponencial, como pode ser constatado em TEICHOLZ (1985). Observou-se um crescimento médio de 35% ao ano, no número de estações de trabalho (workstations) comercializadas pelos EUA, para o período de 1979 a 1989.

Neste estágio, o objetivo principal em automatizar o processo de desenvolvimento de produtos, foi o de aumentar a sua produtividade, e conseqüentemente, reduzir custos de projeto. Para isto procurou-se identificar e automatizar as etapas mais lentas do processo. Das atividades que o computador pode realizar com vantagens sobre o processo manual, destacam-se a produção e modificação dos documentos de projeto.

Alguns índices de produtividade obtidos pelo CAD, e comparados com o processo manual, são apresentados na tabela abaixo, extraída de TEICHOLZ (1985). Observa-se que o desempenho do CAD, naquela época, já representava em média, uma economia de 40% do tempo; o que por si só justificou o investimento em automação para um grande número de aplicações.

Aplicação	Produtividade Média
Documentação do projeto	4,4 : 1
Desenho Esquemático	4,2 : 1
Detalhamento mecânico	2,4 : 1
Desenho de Circuitos Integrados	18,4 : 1
Leiaute industrial	3,4 : 1
Modelagem estrutural	3,1 : 1
Desenhos de Eng. Civil	3,0 : 1
Circuito impresso	3,3 : 1
Desenho de Mapeamento	5,1 : 1

*Tabela 1 - Ganhos de produtividade com uso do CAD em tarefas de projeto*

A importância que a automação de projetos alcançou, e a expectativa criada com as mudanças que ela provocou, levaram diversos pesquisadores a estudar os aspectos sociais e organizacionais ligados a esta tecnologia.

MAJCHRZAK; SALZMAN (1989) mostraram que 75% das tecnologias avançadas de manufatura, implementadas nas empresas americanas, não atingiram os benefícios esperados. Sugerem um cuidado maior no planejamento da mudança tecnológica, enfatizando a organização do trabalho, e não somente os meios e as técnicas empregadas. Observam que os maiores benefícios da tecnologia de automação decorrem de uma interação positiva entre fatores da situação e fatores socio-organizacionais. Entende-se por fatores da situação de um sistema de projeto automatizado: as ferramentas empregadas, o mercado de trabalho e a aplicação pretendida; e por fatores socio-organizacionais: a vivência de projeto da equipe, as habilidades de cada indivíduo, o processo de implementação e a cultura da organização.

LIKER; FLEISCHER (1989) observam que, apesar da implantação do CAD, os projetistas passam apenas uma parte do seu tempo nos postos de trabalho. O restante do dia continua sendo gasto na troca de informações com outros projetistas, ou na procura e recuperação de dados importantes para o seu trabalho. Tais atividades não foram alteradas neste estágio da automação de projetos.

### **2.2.3 A fase do CIM**

Uma vez estabelecidas as chamadas ilhas de automação, o processo de desenvolvimento de produtos começa a rever a forma com que os projetos são conduzidos, na procura por métodos que maximizem a produtividade do processo como um todo. O objetivo principal da automação de projetos passa agora a ser o de aumentar a produtividade de todo o processo, e não só reduzir o tempo de algumas atividades da empresa. Procura-se aumentar a eficiência de todo o processo produtivo, integrando-o com uso do computador em um movimento chamado de CIM (Computer Integrated Manufacturing), iniciando uma nova fase na automação de projetos (JAIKUMAR; 1993)

Diversos estudos levam à conclusão de que a automação de projetos não pode estar dissociada das demais tecnologias presentes em uma organização, passa-se a considerá-la como parte de uma estratégia global de negócios da empresa. As bases do CIM são formadas pela integração do planejamento

da produção com a automação de projetos e com a automação da manufatura.

O CIM procura aumentar a produtividade global da empresa, integrando as ilhas de automação deixadas na fase do CAD/CAM. A ênfase é maior na etapa de produção, do que no projeto. O CIM é uma técnica que tem sido aplicada com sucesso pelas indústrias de bens de consumo mecânicos e electro-eletrônicos (KAPLAN; 1993). A partir dele se desenvolveram técnicas específicas de projeto, com vistas à facilitar a fabricação e montagem (PENNINO; POTECHIN, 1993).

Formam-se ambientes integrados de projeto pela união de ferramentas padronizadas de trabalho (GRAHAM, 1991). Procuram-se formatos e protocolos de comunicação padronizados para viabilizar esta integração, permitindo ao projeto poder fluir por entre as ferramentas de software. Receberam incentivo, durante esta fase, os conceitos de sistemas abertos, que buscam a portabilidade e a interoperabilidade entre aplicações e dados, permitindo a criação de sistemas de projeto independentes de um único fornecedor de hardware ou software. Entre os esforços de normatização destaca-se a norma ISO 10303, que define o padrão STEP para integração da informação dos produtos entre sistemas de projeto. (SCALA; VAQUERO, 1995)

Diversas organizações internacionais, além da ISO, criaram normas reguladoras para os segmentos específicos da

indústria de informática que atende a automação de projeto (DIGITAL, 1991)

#### **2.2.4 A fase da empresa virtual**

Nos anos 90, inicia-se uma nova fase na automação de projetos, motivada pelo aumento, sempre crescente, das exigências dos consumidores. A qualidade do produto passa a ser característica mais importante a ser atendida pelo projeto. Não basta apenas produzir em quantidade e com baixos custos, são fundamentais para o sucesso de um produto a sua aceitabilidade, funcionalidade e o prazo no qual ele chega ao mercado.

O conceito de qualidade utilizado aqui seguem as propostas descritas por FALCONI (1992). Qualidade é entendida a satisfação total do consumidor, obtida não num único aspecto do produto, mas sim na sua totalidade, incluindo a funcionalidade, o preço e a rápida disponibilidade do produto ao mercado. O processo de desenvolvimento de produtos deve rever constantemente a forma com que os projetos são conduzidos, na procura por métodos que garantam e maximizem estas qualidades (HAMMER; CHAMPY, 1993).

Levada ao extremo, a busca pela excelência na qualidade é obtida com uma nova organização das empresas, que prioriza o projeto à produção, com maior delegação de responsabilidades e valorização do trabalho das equipes de

projeto. Os processos tradicionais, formais e departamentalizados, dão lugar à chamada equipe virtual (do inglês: *virtual team*, SPRAGUE et al.; 1991), que constituem a base da empresa virtual (do inglês: *Virtual Enterprises*, REDDY et al.; 1993). O computador e as tecnologias de comunicação criam laços virtuais de ligação entre os envolvidos com o desenvolvimento de um produto. Realizado o projeto, os documentos resultantes são transferidos eletronicamente, para as instalações flexíveis de manufatura para ser produzido, que já estão bem estabelecidas desde a fase do CIM.

O produto deve ser projetado em equipe, com a participação de todos os envolvidos no processo, assim terá mais chance de ser projetado corretamente, não incorrendo em erros de fabricação, montagem ou uso. Além de ser barato e chegar ao mercado rapidamente. Tais idéias foram inicialmente reunidas pelo IDA dos EUA, em 1986 sob o título de engenharia simultânea (CARTER; BAKER, 1991), em uma metodologia de projeto proposta para redução dos elevados custos de desenvolvimento de projetos na indústria de defesa norte-americana. Na engenharia simultânea o projeto não deve descrever somente como devem ser os produtos, mas também como são os processos para sua produção.

Alguns estudos avançados sobre este assunto foram liderados pelo DARPA (Defende Advanced Research Projects Agency), que em 1988 lançou o DICE (DARPA's Initiative on Concurrent

Engineering). Este programa de trabalho, formado por um consórcio de diversas empresas de informática e empresas usuárias objetivou pesquisar, desenvolver ferramentas, validar e transferir tecnologias que permitam a prática da engenharia simultânea pela indústria norte-americana (REDDY et al., 1991).

A Universidade de West-Virgínia, atuando como o braço acadêmico do projeto, implantou o CERC (Concurrent Engineering Research Center). Neste centro foram desenvolvidos cursos e programas de divulgação das propostas apresentadas pelos participantes do DICE. Os resultados, além de uma série de trabalhos publicados, foram colocados à disposição da comunidade científica através da Internet (CERC; 1993)

A engenharia simultânea impõe grandes mudanças ao processo tradicional de desenvolvimento de produtos. Uma grande ênfase do estudo é dada à fase de projeto, em complemento ao CIM, que enfatiza a fase de produção. O trabalho em equipe e a integração das diversas especialidades envolvidas, tem um importante papel nesta nova sistemática de trabalho. Procura-se utilizar os recursos computacionais para aumentar a comunicação na equipe, mais do que para automatizar o processo de criação e desenho do produto. O CAD passa a ser uma poderosa ferramenta de comunicação entre os componentes de uma equipe de projeto, como destaca ROSENBLAT (1991).

Sistemas computacionais para compartilhar e gerenciar as informações de projeto tornam-se importantes para que os projetistas possam desempenhar todas as suas atividades através do computador. Com este auxílio o projetista pode recuperar informações, processá-las e comunicar seus resultados. Cresce ainda mais a necessidade de padrões para a integração das atividades de engenharia no computador.

Normalmente, os dados de projeto são encontrados em diferentes formatos, como os modelos de CAD, os bancos de dados, as bases de conhecimento, entre outros. A diversidade das ferramentas, e a multiplicidade de passos para acesso à informação criam sérias barreiras à integração. MORGENSTERN (1992) alerta para a importância do acesso livre à informação no sucesso de um ambiente de engenharia simultânea. Propõe que se desenvolva uma interface de alto nível, com recursos de multimídia, para facilitar este acesso.

A seleção e a integração dos sistemas computacionais para apoiar o trabalho das equipes virtuais torna-se uma atividade muito complexa, da qual depende o sucesso da automação de projetos. Deve-se aliar a especialização do software utilizado pelos diferentes elementos da equipe, com a facilidade de comunicação dos dados entre eles. O engenheiro de software é convocado para viabilizar esta integração e criar os novos ambientes de trabalho.

CUTKOSKY et al. (1993) examinam os aspectos tecnológicos, da construção de sistemas computacionais distribuídos, para engenharia simultânea. Realizam um experimento de projeto, análise, teste e fabricação totalmente automatizado, de uma placa de circuito eletrônico. O trabalho foi realizado em conjunto com diversas Universidades e empresas norte-americanas. A abordagem seguida procurou integrar ferramentas de software e hardware de diferentes fornecedores, ainda que não tenham sido previamente concebidas para operarem em conjunto. Cada participante criou uma camada de software padronizado que envolveu a sua ferramenta original e permitiu a troca da informação entre eles, através da Internet. Este experimento encontra-se na fronteira tecnológica da automação de projetos, e permitiu identificar diversos problemas de integração.

Os desafios gerenciais encontrados para implantação de um sistema de automação foram classificados por ADLER (1989), segundo os 5 níveis crescentes do aprendizado de uma organização: habilidades, processo, estrutura, estratégia e cultura. Vencidos os desafios de um nível, obtém-se apenas parte dos potenciais benefícios da tecnologia para a organização, mas abre-se o caminho para novos desafios, encontrados pela tentativa de incorporar a tecnologia nos níveis superiores. Esta classificação coincide com as etapas aqui propostas para a evolução da automação. Venceu-se a fase da habilidade, dos primórdios da computação

gráfica para o CAD/CAM. A automação de projetos chega ao nível de processo com o CIM. A utilização da automação de projetos como estratégia da empresa se dá com o movimento da empresa virtual. Após a consolidação desta tecnologia chega-se a uma cultura de projeto automatizado.

#### **2.2.5 A automação de projetos no Brasil**

As empresas brasileiras acompanharam o processo internacional da automação de projeto, atravessando fases semelhantes na sua evolução. A Sociedade Brasileira de Comando Numérico e Automatização Industrial, entidade civil representante deste segmento de indústrias, registrou, o crescimento do número de postos de CAD comercializados no país, no período de 1986 a 1992, ainda sobre a Lei de Reserva do Mercado de informática, numa média de 40% ao ano (SOBRACON, 1992).

As empresas de consultoria em engenharia foram estudadas por PLONSKI (1989), que observou alguns insucessos na implantação do CAD. Tais resultados foram atribuídos à falta de um entendimento completo dos potenciais benefícios do CAD, encarado pela maioria apenas como um meio de automatizar o desenho; e também, a um plano de implantação deficiente, que não posiciona corretamente o homem como centro do processo, subdimensionando as necessidades de treinamento e aculturação.

A EMBRAER (Empresa Brasileira de Aeronáutica) tem sido uma empresa pioneira na automação de projetos. Para atender ao mercado mundial, esta empresa procurou empregar o que há de mais moderno em ferramentas computacionais de projeto como seu diferencial estratégico. Iniciando nos anos 70 a implantação de sistemas para desenho em computador, e sistemas de auxílio à manufatura. A empresa deu um salto qualitativo em 1987, quando da implantação de mais de 100 estações de trabalho de CAD; neste momento a totalidade dos projetos da empresa passaram a ser realizados pelo computador (BUONO, 1995).

A indústria automobilística tem sido também uma grande usuária de sistemas de projeto automatizado, normalmente seguindo as linhas de trabalho da suas respectivas empresas matriz. A exigência por peças de maior qualidade também levou o computador ao setor de autopeças, que em conjunto com as empresas montadoras de veículos lideram o mercado de automação de projetos no Brasil. (D'ISSY, 1995).

A indústria de defesa no Brasil também é uma impulsionadora desta tecnologia. Não apenas na EMBRAER, onde participa a Força Aérea Brasileira, mas também no CTMSP, órgão brasileiro ligado ao Ministério da Marinha; o CAD é empregado desde 1986 em projetos ligados ao desenvolvimento de reatores nucleares para propulsão naval. Os primeiros sistemas se limitavam a apoiar a produção de desenhos, hoje

ajudam a criar modelos gráficos tridimensionais das complexas instalações projetadas.

A preocupação com a automação de projeto nas empresas brasileiras não se restringiu às grandes corporações. O resultado de visitas à empresas de pequeno e médio porte apresentado por CLEMENTE et al. (1992) mostrou que na sua maioria, apesar de existentes, os sistemas de CAD/CAM estavam sendo deixados em segundo plano, persistindo os métodos tradicionais de trabalho. Identificaram-se como causas deste problema a falta de treinamento e suporte técnico adequado ao porte das empresas, que não conseguem vencer a resistência dos projetistas na adoção desta tecnologia.

No Brasil a aplicação das técnicas de Engenharia Simultânea é mais recente que nos demais países industrializados, mas vem sendo motivada pela globalização da nossa economia (IANNI, 1994). Os trabalhos sobre o assunto relatando experiências sobre projetos desenvolvidos com técnicas de Engenharia Simultânea passam a ser encontrados em diversos Congressos da área nos últimos anos, como o CONGRESSO INTERNACIONAL DE COMPUTAÇÃO GRÁFICA (1994 e 1995) e o CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL (1994).

### **2.3 Modelagem do processo de projeto em engenharia**

O entendimento do processo de projeto é condição necessária para uma automação correta. O engenheiro de software deve

conhecer em detalhes o processo, para poder automatizá-lo. Diversas técnicas ajudam a compreender e a modelar as atividades desempenhadas pelos engenheiros projetistas, no desenvolvimento de um produto. Cada técnica de análise está associada a um estilo de modelagem do processo. Identificando-se os aspectos comuns de cada modelo é possível se construir um modelo geral do processo de projeto em engenharia.

### **2.3.1 Técnicas para modelagem do processo**

Observa-se uma equivalência entre as várias técnicas utilizadas para representar um processo de projeto em engenharia e as técnicas utilizadas pela engenharia de software.

SHINA (1991) propõe o uso da notação dos Diagramas de Fluxo de Dados, e demais técnicas da engenharia de software, como apoio ao entendimento das relações envolvidas no desenvolvimento de um produto, e para a documentação do processo de trabalho. Destaca que apesar da simplicidade da notação, estes diagramas possuem uma grande riqueza de detalhes de representação.

SCHWAB et al. (1993) destacam a importância da modelagem do processo de trabalho, principalmente quando os sistemas de informação serão responsáveis por integrar toda a organização. Os modelos devem apresentar um nível de detalhe onde se possa distinguir pequenas diferenças entre

os processos, permitindo a análise e o refinamento dos sistemas.

DE WYZE (1991) propõe o uso do IDEF0 (Integrated Definition), uma notação gráfica desenvolvida pela Força Aérea dos EUA no projeto ICAM (Integrated Computer-Aided Manufacturing), para descrever um processo de manufatura. O IDEF0 resulta em um modelo semelhante ao de um fluxograma. Uma família de métodos derivaram dele, formando um conjunto de técnicas integradas para representar processos e dados. Os modelos IDEF3 se propõem a descrever sistemas através de um modelo de estados e seus eventos, enquanto os modelos IDEF4 descrevem em uma representação orientada a objetos. Os trabalhos de MAYER (1992a e 1992b) analisam esta metodologia.

KARINTHI et al. (1992) propõem o uso de uma linguagem de definição de dados chamada EXPRESS, desenvolvida com base no padrão STEP, com objetivo de modelar um empreendimento complexo e integrar as diferentes bases de dados usadas pela engenharia. A linguagem EXPRESS, é orientada a objetos e permite a definição de classes relativas ao processo de projeto. Estas classes são armazenadas em um banco de dados que permite uma visão completa do empreendimento. SWAB et al. (1993) também destacam a importância do padrão STEP na automação dos processos de projeto, prevendo um considerável impacto sobre os sistemas de desenvolvimento de produtos e no planejamento da manufatura, quando este

padrão estiver difundido pela indústria de software. SCALA; VAQUERO (1995) descrevem a aplicação da linguagem EXPRESS na integração dos sistemas de informação para a manufatura integrada, detalhando as vantagens desta utilização.

### **2.3.2 Modelos do processo de projeto de engenharia**

Diversos autores desenvolveram seus próprios modelos do processo de projeto. Destacam-se os trabalhos do CIM-OSA (ROZENFELD et al.; 1992), CARTER; BAKER (1991), REDDY et al. (1993) e CUTKOSKY et al. (1993).

ROZENFELD et al. (1992) analisam a necessidade da integração da automação com o planejamento estratégico das empresas, propondo o desenvolvimento de um modelo único para ambas as finalidades. O modelo da empresa organiza as informações sobre o funcionamento da organização, permitindo o entendimento dos processos internos e externos, e atendendo ao mesmo tempo à integração da manufatura. Apresentam uma revisão das metodologias desenvolvidas até então, destacando o modelo proposto pelo CIM-OSA.

O CIM-OSA (Open Systems Architecture), desenvolvido em conjunto com o programa europeu ESPRIT, divide um ambiente de manufatura em 3 níveis básicos de integração (CIM-OSA, 1996):

- a) integração de sistemas físicos de comunicação, onde se integram as informações entre os postos de trabalho, ferramentas de projeto e meios de produção;
- b) integração de aplicações, que promove o compartilhamento dos modelos utilizados para o desenvolvimento de um produto, e das informações disponíveis nos sistemas de informação;
- c) integração do negócio: que integra as funções gerenciais e administrativas da empresa como finanças, marketing e planejamento com a produção.

CARTER; BAKER (1991) propõem para os ambientes de desenvolvimento baseados na engenharia simultânea, uma estrutura segundo 4 direções principais: organização, infra-estrutura de comunicação, requisitos do produto e desenvolvimento do projeto. Em cada direção um conjunto de fatores-chave decide o sucesso da automação. A tabela abaixo resume este modelo.

<b>Direção</b>	<b>Atividades</b>	<b>Fatores-chave</b>
Requisitos do Produto	Representar o produto e as necessidades dos clientes	<ul style="list-style-type: none"> <li>• Definição dos requisitos</li> <li>• metodologia de planejamento</li> <li>• escopo do planejamento</li> <li>• validação dos requisitos</li> <li>• padronização</li> </ul>
Organização	gerenciamento planejamento formação das equipes	<ul style="list-style-type: none"> <li>• integração das equipes</li> <li>• delegação de responsabilidades</li> <li>• educação continuada e</li> <li>• suporte computacional</li> </ul>

Desenvolvimento	Desenvolvimento dos produtos	<ul style="list-style-type: none"> <li>• subdivisão dos componentes</li> <li>• metodologia de projeto</li> <li>• melhoria contínua</li> <li>• otimização</li> </ul>
Infra-estrutura de comunicação	Troca de informações	<ul style="list-style-type: none"> <li>• gerenciamento da comunicação</li> <li>• gerenciamento dos produtos</li> <li>• gerenciamento da documentação</li> </ul>

*Tabela 2 - Modelo do ambiente de engenharia simultânea (CARTER; BAKER, 1991)*

REDDY et al. (1993) descrevem, com detalhes, um ambiente de projeto colaborativo ideal. Este ambiente é construído pela integração de diversas tecnologias computacionais, organizadas em uma arquitetura de camadas, apresentadas na figura 3:



*Figura 3- Modelo de camadas da organização de uma equipe virtual*

a) Camada de atividade, que representa as atividades desempenhadas pela equipe de projeto, envolvidas em um plano de melhoria contínua, como sugerem os programas de qualidade total;

b) Camada de transação, onde são desenvolvidas as atividades desempenhadas pelos integrantes de uma equipe de projeto, que desempenha as atividades descritas na tabela abaixo:

<b>Atividade</b>	<b>Descrição</b>
pesquisar	procurando em servidores de informações os dados necessários ao seu trabalho
calcular	acrescendo informação ao modelo do projeto, através de ferramentas específicas de acesso ao modelo
comunicar	compartilhando suas decisões de projeto com os demais membros da equipe
negociar	influenciando nas decisões dos outros membros da equipe
decidir	escolhendo, através de ferramentas de apoio à decisão, as melhores opções de projeto
arquivar	mantendo as informações de projeto disponíveis, e permitindo o acompanhamento da evolução do projeto

Tabela 3 - Atividades da camada de transação do modelo de REDDY et al. (1993)

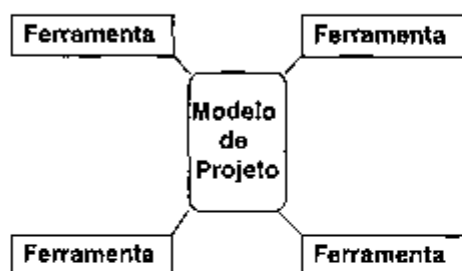
c) Camada de Serviços : uma variedade de serviços que suportam as atividades fundamentais dos membros da equipe, organizados como mostra a tabela a seguir:

<b>Serviço</b>	<b>Descrição</b>
Colocação	comunicação entre membros geograficamente dispersos
Informação	compartilhamento de informações armazenadas em formatos heterogêneos pela organização
Coordenação	coordenação de uma equipe multidisciplinar, incluindo visualização comum dos dados, sistemas de apoio à decisão e negociação das soluções
Gerenciamento da História Corporativa	captura da evolução do projeto, importante para preservar o conhecimento de um projeto para auxiliar projetos futuros
Integração	união entre os demais serviços, permitindo a troca de dados entre ferramentas entre a organização

Tabela 4 - Descrição da camada de serviços (REDDY et al., 1993)

- d) Camada do Modelo Corporativo de Informação: disponibiliza a informação para os serviços apresentados acima, criando o chamado modelo de informações da corporação.
- e) Camada de recursos e rede : representa as fundações onde é implementado o ambiente de projeto para a equipe virtual.

Procura-se preservar os investimentos já realizados pelas empresas em sistemas de automação de projetos, mesmo com as dificuldades existentes para integração. Cada ferramenta disponível hoje apresenta uma solução própria para o problema de modelagem e análise, baseando-se em notações, formatos e algoritmos particulares. CUTKOSKY et al. (1993) propõem que a integração de sistemas se dá através do compartilhamento do modelo de projeto, pela interação das ferramentas de projeto, como esquematizado na figura abaixo:



*Figura 4 Interação do modelo com as ferramentas de projeto*

O projeto, segundo CUTKOSKY et al. (1993), é um processo de negociação entre os diversos modelos. As ferramentas de projeto devem ajudar os projetistas a compartilhar o conhecimento e acompanhar a evolução dos requisitos,

necessidades, restrições, compromissos e decisões dos demais projetistas. A maioria dos modelos de projeto usados, por exemplo um arquivo CAD, são representações superficiais, e incapazes de compartilhar toda a informação necessária. Para que os modelos de projeto formem uma base de comunicação do conhecimento entre os diferentes sistemas, CUTKOSKY et al. (1993) propõem que as ferramentas sejam envolvidas por uma camada externa de software formando o que chamou de um agente, como mostra a Figura 5. Os agentes são módulos de software que se comunicam entre si, através de outros módulos, chamados de facilitadores, e que permitem o processamento de um ambiente integrado de projeto.

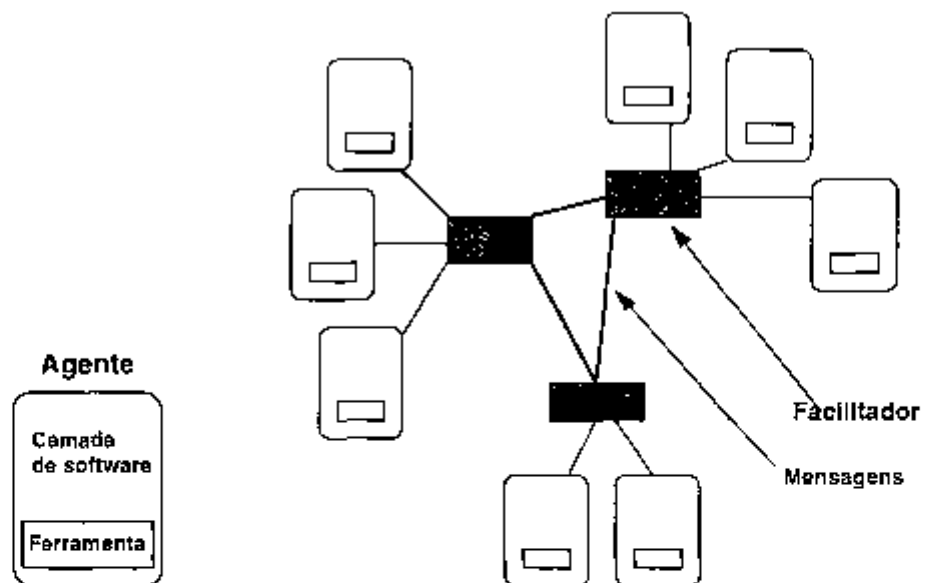


Figura 5 - Estrutura de integração de projeto (CUTKOSKY et al., 1992)

### 2.3.3 Arquitetura geral de um sistema de projeto em engenharia

Pode-se integrar as visões apresentadas por CARTER; BAKER (1991), ROZENFELD et al. (1992), REDDY et al. (1993) e CUTKOSKY et al. (1993), criando um modelo geral para a arquitetura de um sistema de projeto, como foi apresentado por MARTINI; DEBONI (1994a). Este modelo, organizado em camadas, encontra-se esquematizado na figura 6. Cada camada traduz uma visão diferente do processo de projeto: gerencial, do projetista e da infra-estrutura. Cada camada constitui um problema específico de automação, sendo que as camadas superiores impõem requisitos para as camadas inferiores que por sua vez, oferecem serviços às camadas superiores. O produto, pode ser considerado com uma quarta camada, que estabelece os requisitos e se serve das demais.

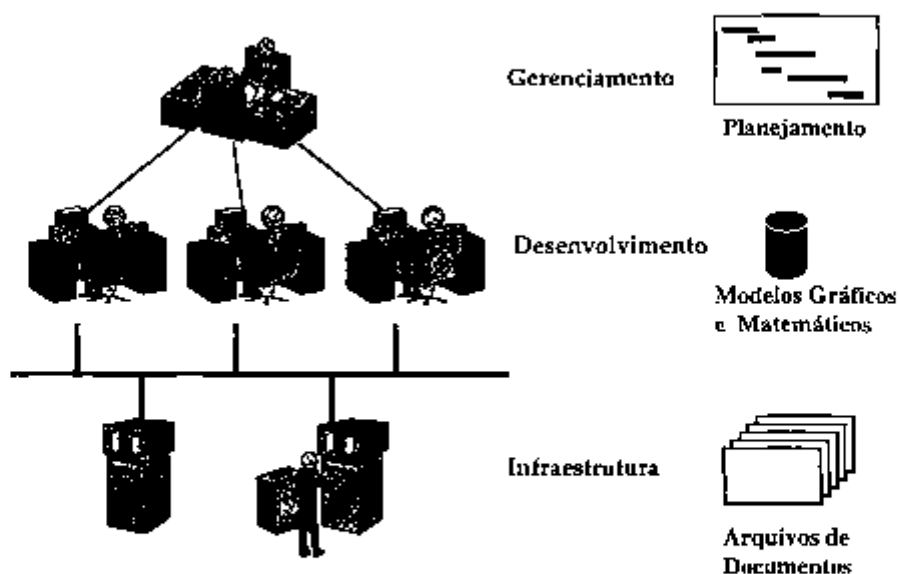


Figura 6 - Modelo de organização do ambiente de projeto

A base para o desenvolvimento integrado de projetos, neste modelo, é a formação de equipes de projeto. Em postos de trabalho dispersos geograficamente, a equipe de projeto compartilha, eletronicamente, modelos matemáticos de análise e modelos gráficos do produto que está sendo projetado. O gerente de projetos organiza as equipes, delegando à elas a responsabilidade pela condução de parte do projeto, ele controla a qualidade do resultado final, e o cumprimento de um plano de trabalho. Uma infra-estrutura computacional garante a integridade da informação no sistema, e gerencia os documentos eletrônicos produzidos pelos projetistas. A tabela seguinte apresenta os serviços associados a cada camada e uma breve definição.

Camada	Função	Definição da função na arquitetura geral
gerenciamento	planejar	dividir o trabalho em atividades, com base nos requisitos do produto, procurando minimizar as interferências
	delegar	criar uma equipe e distribuir as atividades entre elas, um evento caracteriza grupos de atividades
	controlar	observar a execução de um atividades no tocante a prazo, custo e atendimento a requisitos
	realimentar	agir sobre a execução de uma atividade no sentido de corrigir desvios e incentivar bons resultados (remuneração)
Desenvolvimento	pesquisar	procurar informações e dados necessários para o trabalho de uma equipe de projeto
	modelar	acrescentar informações de projeto aos modelos, criando novas soluções ou analisando soluções propostas.

	decidir	optar por uma das soluções propostas no sentido de otimizar os requisitos, tomar decisões de projeto
	negociar	influenciar nas decisões de projeto de outras equipes ou de outros membros da equipe. Assumindo soluções de compromisso
Infra-estrutura	arquivar	manter as informações de projeto disponíveis às equipes de projeto, e permitindo o acompanhamento da evolução do projeto
	comunicar	compartilhar as decisões de projeto com os demais membros da equipe
	recuperar	resgatar os dados e informações necessárias ao trabalho
	capturar	entrar informações de projeto no sistema, converter outros documentos em documentos eletrônicos
	imprimir	sair informações de projeto, converter os documentos produzidos em outras mídias
	traduzir	converter o formato entre documentos, permitir a troca da informação
	facilitar	incluir as operações que não estão diretamente ligadas ao projeto, mas viabilizam, simplificam ou agilizam as outras operações.
	treinar	garantir a continuidade da evolução do sistema, aumentando e mantendo as especializações

Tabela 5 - Lista de funções da arquitetura geral de um sistema de projetos automatizado

### 2.3.3.1 Camada de gerenciamento

O planejamento do projeto, a formação e a delegação de responsabilidade às equipes de desenvolvimento são

atividades associadas à esta camada do modelo, desempenhadas pelo gerente de projeto.

O planejamento do trabalho, em um ambiente automatizado de projeto, deve considerar a metodologia de controle do projeto, assim como as facilidades oferecidas pelos novos ambientes de desenvolvimento. Do planejamento tradicional detalhado e subdividido em fases, para um sistema de projeto multidisciplinar automatizado se procura subdividir o produto em partes, definir requisitos do projeto, as interfaces e as restrições que cabem a cada uma, e delegar responsabilidade às equipes de projeto.

MACKEY; CARTER (1994) propõem que um projeto de engenharia seja planejado e controlado com base em eventos distribuídos entre as equipes. Cada equipe deve informar, periodicamente, através de relatórios de acompanhamento, quanto o evento evoluiu em uma escala de 0 a 100%. Métodos baseados em homens-hora, fases, ou documentos emitidos provaram-se imprecisos ou ineficientes para medir o andamento do projeto no ambiente automatizado.

Os eventos devem ser selecionados para poderem ser executados por equipes diferentes, admitindo simultaneidade no trabalho das equipes, desde que minimizadas as interferências e interfaces entre elas. KUSIAK; WANG (1993) apresentam um método para decompor as atividades planejadas de modo a minimizar as interferências e aumentar a possibilidade de paralelismo durante o desenvolvimento.

### **2.3.3.2 Camada de desenvolvimento**

A camada de desenvolvimento está associada às atividades das equipes de desenvolvimento do produto. A captura dos requisitos de projeto, os padrões a serem adotados, o processo de criação, análise, validação e otimização do projeto, são aqui realizadas.

A automação de projeto visa a aumentar a qualidade do produto e abreviar a sua chegada ao mercado. O envolvimento do cliente, durante todo o ciclo de projeto, é tão importante, que para garantir que os requisitos sejam satisfeitos as equipes de projeto podem incluir representantes dos próprios clientes como seus membros. Estes métodos diferem dos métodos tradicionais de trabalho compartimentalizados, pois envolvem uma grande dose de comunicação e negociação.

Os requisitos dos cliente devem ser transformados em requisitos para o produto, para que haja uma rastreabilidade deles, nas propriedades do produto. Os requisitos servem também para validar o projeto servindo de critério para a tomada de decisão.

### **2.3.3.3 Camada de infra-estrutura**

A camada da infra-estrutura cuida da informação de projeto organizada no seu nível mais elementar: o documento eletrônico. Esta camada segue o ciclo da gerência de documentos em um ambiente automatizado de engenharia

discutido por MARTINI; DEBONI (1994b). Tratam-se dos problemas de captura, tradução de formatos, arquivamento, circulação, recuperação e impressão dos documentos de projeto.

A infra-estrutura forma o substrato para o desenvolvimento do produto, e define como as informações são trocadas entre os componentes das equipes e deles com o gerenciamento. O trabalho em equipe exige grande dose de comunicação e a infra-estrutura é a responsável por facilitar o trabalho do grupo. Nesta camada definem-se os meios e formatos para a troca de informação.

#### **2.4 Produtos de software para automação de projetos em engenharia**

A evolução da automação de projetos foi acompanhada pela evolução das ferramentas computacionais que a implementam. Criou-se um mercado específico para produtos de software de aplicação na engenharia. TEICHOLZ (1985) já apontava, nas primeiras fases da automação de projetos, algumas dezenas de produtos disponíveis. Atualmente, os catálogos dos principais fornecedores de hardware, como SILICON (1994), INTERGRAPH (1996) e SUN (1994) registram milhares de produtos de aplicação nas diversas áreas de especialização da engenharia.

Os primeiros sistemas computacionais se apoiavam em grandes computadores, com periféricos especiais e software

dedicado, em geral, comercializados pelo próprio fabricante. O usuário não era obrigado a conhecer profundamente os assuntos de computação, bastando selecionar um bom fornecedor, que este apresentaria uma solução completa para o seu problema de automação. Hoje um único fornecedor não é mais capaz de atender a todos os aspectos do problema de automação. A solução é formada pela união de vários programas, com origem em diferentes fornecedores.

A construção de sistemas pela compra de pacotes de software prontos é uma das aplicações possíveis da reutilização de software, como observa PUJATTI (1994). Define-se aqui reutilização de software como o uso de programas de computador previamente desenvolvidos e testados, em novas situações, que em conjunto com outros programas, formam um novo sistema. Com este tipo de re-aproveitamento não é mais necessário o teste individual das partes, restando testar apenas a integração entre elas.

A reutilização sistemática de porções de software resulta em um menor esforço de projeto, produtos finais de maior qualidade, menor custo e maior facilidade de manutenção (DAVIS; 1994). Observa-se que normalmente os custos de aquisição de um software são inferiores aos do seu desenvolvimento. (SHRIVER, 1987 apud PUJATTI, 1994).

Destacam-se duas técnicas para a reutilização de software: composição e a geração. Na composição o software é o

resultado da união de componentes básicos; enquanto na geração os programas são recriados a partir de modelos estruturais.

PRIETO-DIAZ; FREEMAN (1987) afirmam que a reutilização de software por composição necessita:

- a) disponibilidade de um grande número de componentes de software confiáveis,
- b) a existência de um mecanismo de agregação das partes para a formação do todo, e
- c) que as partes sejam catalogadas, podendo ser facilmente localizadas e avaliadas.

Verifica-se que os sistemas computacionais para apoio a projetos de engenharia atendem aos requisitos para composição de software de PRIETO-DIAZ; FREEMAN (1987), pois se dispõe de um grande número de programas, que podem ser agregados na formação de um sistema. Analisam-se a seguir os mecanismos disponíveis para a integração destes produtos em um sistema, e um critério para classificação destes programas que permita uma localização e avaliação.

#### **2.4.1 Integração de produtos de software**

A operação integrada dos produtos de software em um ambiente de processamento compartilhado pede que sejam satisfeitas duas condições:

a) um sistema operacional comum onde os produtos possam coexistir e cooperar, e

b) um mecanismo eficaz para troca de informação entre eles.

Ambas as condições dependem do atendimento a normas e padrões de integração por parte dos produtos. A portabilidade de dados e aplicações e a interoperabilidade são alguns dos benefícios dos sistemas que atendem aos padrões de sistemas abertos (DIGITAL, 1991).

A procura por padrões para troca de informações entre produtos de software sempre foi um assunto de grande importância para a automação de projetos, e motivo de um grande esforço de normatização, resumido a seguir.

#### **2.4.2 Sistemas operacionais para integração de software**

De um modo geral, todo software é uma extensão do sistema operacional, e adiciona uma nova funcionalidade a ele sem, necessariamente, reconstruí-lo. Esta propriedade é conhecida como extensibilidade de software. Define-se extensibilidade como a propriedade de adicionar novas funções, ou renovar partes de um programa sem a necessidade de recriar o que já está operando corretamente. (UDEL, 1994).

Um dos primeiros sistemas operacionais a explorar esta propriedade foi o sistema UNIX, que deu origem ao padrão POSIX, normatizado pela ISO 1003.0, e que é atendido pela maioria dos sistemas operacionais das estações de trabalho

com processadores RISC. A interface padronizada para sistemas abertos é o OSF/Motif, que deve ser atendida pelos produtos de software que desejem se integrar neste ambiente (BERSON, 1992).

Apesar de não obedecer a um padrão *de jure* deve-se considerar como um padrão de mercado o sistema operacional MS-WINDOWS, dada a grande quantidade de software disponível para esta plataforma, integrados aos processadores da linha INTEL 80X86 amplia seu espectro de aplicação acompanhando o crescimento do poder de processamento destes processadores (MILLS; BECKERT, 1991).

A extensibilidade é um dos benefícios prometidos pela programação orientada por objetos aplicada na construção de sistemas operacionais. Não é necessário conhecer o objeto que será processado até o momento de processá-lo, o que permite que um código velho acione um código novo (FOUNTAIN; SZYPERSKI, 1994).

Associados a ambos os sistemas operacionais padrões, duas correntes competem no mercado para oferecer uma extensibilidade completa ao sistema operacional, pela distribuição de objetos: o COM da Microsoft, e o SOM desenvolvido pela IBM, e apoiado pela Apple, SUN entre outros.

Os dois modelos são significativamente diferentes. O SOM é independente de linguagem e suporta completamente a

programação orientada por objetos, enquanto o COM suporta apenas dos princípios da orientação por objetos.

UDELL (1994) analisa o crescimento da indústria de componentes de software, popularizados por ferramentas de desenvolvimento de aplicações em ambiente padrão Windows. Estes componentes não são rigorosamente objetos, porque atendem, unicamente, ao encapsulamento. Porém oferecem uma solução para suplantar os obstáculos da distribuição de objetos de software na forma binária, provendo mecanismos para troca de componentes para construção de produtos de software.

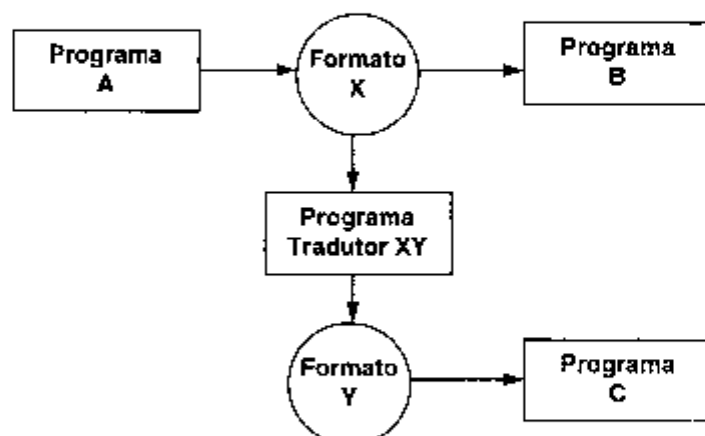
O OpenDoc é um padrão para a composição de documentos entre plataformas definido no SOM, que vai além da composição de documentos definido uma arquitetura para componentes de software. Ele se baseia em uma tecnologia para inclusão de componentes (assim como o OLE), um formato para arquivo de componentes, uma arquitetura de linguagem neutra, e um mecanismo de ligação e troca de componentes em tempo de processamento (ORFALI; HARLEY, 1995).

Observa-se que estão se criando as bases para uma indústria de componentes de software (Componentware), sendo aplicada tanto nos sistemas operacionais como nas próprias aplicações. Esta indústria irá aumentar as possibilidades de reutilização de software pois as aplicações serão criadas com base em componentes padrões que poderão ser facilmente integrados.

Conclui-se que é possível construir um sistema de processamento integrando produtos de software, desde que estes produtos atendam a um destes padrões emergentes. Padronizando-se a plataforma de sistema operacional a ser utilizado, garante-se que os programas poderão coexistir e compartilhar informações. A adoção de padrões de sistemas abertos traz um ônus de performance e custo, por vezes alto demais, casos onde é aceitável a utilização de padrões de mercado ou sistemas proprietários.

#### **2.4.3 Padrões para troca de informação entre produtos de software**

A comunicação entre os produtos de software pode ser estabelecida através do compartilhamento de um arquivo em formato padronizado, como mostra a figura abaixo. Para haver a troca de informações é necessário que os dois produtos obedeam a um determinado formato de arquivo, ou que os formatos se diferentes, possam ser traduzidos.



*Figura 7 - Integração de ferramentas pela troca de arquivos*

Diversas iniciativas de organismos de padronização disputam com as empresas produtoras de software a tentativa de estabelecer e manter atualizados os padrões para a troca de arquivos, e a comunicação eletrônica de informações de engenharia.

Dos muitos esforços para padronizar a troca de informações entre produtos de software para área de engenharia, destacam-se:

- IGES - INITIAL GRAPHICS EXCHANGE SPECIFICATION e
- STEP - STANDART FOR EXCHANGE OF PRODUCT DATA;
- DXF - DATA EXCHANGE FORMAT.
- CALS - COMPUTER-AIDED ACQUISITION AND LOGISTIC SUPORT;

que serão descritos a seguir.

Um padrão bastante difundido pela indústria de software para engenharia que permite a troca de informações gráficas é o formato IGES. O IGES padroniza o arquivamento de desenhos e modelos tridimensionais, quanto à geometria das formas, em um formato independente dos fornecedores software, controlado nos EUA pelo NITS (DIGITAL, 1991)

Como uma evolução do IGES, o padrão STEP foi desenvolvido por um consorcio independente de empresas, e referendado posteriormente pela norma ISO 10303. O STEP objetiva criar um mecanismo neutro para troca de informações sobre produtos, tanto nas suas características geométricas, como

nos dados técnicos do produto. Trata-se de um padrão complexo onde a linguagem EXPRESS é usada para descrever os dados associados aos produtos (SCALA; VAQUERO, 1995).

Observa-se um aumento gradativo na adoção do padrão STEP pela indústria de software, no entanto um dos padrões mais aceitos pelos fabricantes de software é o formato DXF, propriedade da empresa AUTODESK, que apesar de não ser normatizado é largamente utilizado pelos produtos de software.

Em 1980, o programa CALS, foi criado pelo Departamento de Defesa do governo norte-americano, com objetivo de incentivar a troca eletrônica de informações nos projetos do governo norte-americano. O CALS define os padrões e normas a serem seguidos para o envio de documentos eletrônicos (CALS 1993). O sucesso do programa foi tão grande que ele suplantou os limites do governo dos EUA, sendo adotado por diversas empresas, em âmbito internacional. BRUEN (1992) observa o impacto internacional do CALS, fazendo com que as corporações privadas também reexaminem a participação do gerenciamento de documentos no ciclo de vida dos seus produtos.

Diversas propostas do CALS foram adotadas como padrões de uso internacional, e de ampla aceitação pela indústria de software. É o caso do da linguagem SGML para padronização de documentos textuais que também foi adotada pelo STEP.

Conclui-se que a troca de informações técnicas entre produtos de software é possível se os produtos suportarem alguns dos padrões citados, ou em casos mais específicos, se os programas adotarem outro padrão.

#### **2.4.4 Classificação dos produtos de software**

O objetivo da classificação dos produtos de software é permitir ao engenheiro de software a seleção adequada de um componente, para atender aos requisitos de um sistema de projeto. PRIETO-DIAZ; FREEMAN (1987) propõem classificar as componentes de software por: função que desempenha, forma como desempenha a função e detalhes de implementação.

Neste estudo a classificação de cada produto de software é dividida em quatro itens:

- a) FUNÇÃO: Funções que o software pode desempenhar no sistema, baseado na lista de funções de um sistema automatizado de projeto, relacionada com o modelo geral do sistema;
- b) APLICAÇÃO : Áreas de especialização onde o software pode ser aplicado. A área de especialização segue uma das propostas apresentada para classificação de software discutida em GLASS; VESSEY (1995) e detalhada no anexo A;
- c) PLATAFORMA: Sistemas operacionais e famílias de equipamentos nos quais o software pode ser processado;

d) PADRÕES : Padrões e formatos dos arquivos suportados pelo software, e que estão disponíveis para integração.

O Anexo A especifica um banco de dados de produtos de software, construído segundo os critérios de classificação e integração aqui apresentados. Este banco de dados facilita a localização e avaliação dos produtos de software disponíveis para reutilização.

## **2.5 Conclusões**

Analisando-se a evolução histórica da automação de projetos observa-se uma mudança no seu objetivo principal. Do mero aumento da produtividade, buscado na fase do CAD/CAM, passou-se para uma busca pela excelência na qualidade dos produtos, na fase da empresa virtual. Cada mudança foi acompanhada de alterações nas ferramentas de software disponibilizadas, que procuraram atender aos objetivos da fase anterior e incorporar os requisitos da fase seguinte.

Procurando preservar investimentos já realizados em hardware, software e treinamento, as empresas construíram sistemas computacionais híbridos onde diferentes fases da tecnologia estão, simultaneamente presentes. Nenhuma empresa de projeto moderna prescinde da automação de projetos, condicionando o seu sucesso ao emprego de ferramentas computacionais adequadas no desenvolvimento dos produtos, e da eficiência com que realiza compartilha as informações de projeto. A integração não se restringe

apenas aos modelos de projeto, mas se estende ao planejamento e ao controle da documentação produzida.

A integração dos produtos de software vindos de diferentes fornecedores na formação de um SCAPE é possível desde que se adote padrões adequados para o sistema operacional e para o formato da informação.

### **3. DESCRIÇÃO DO MÉTODO PARA INTEGRAÇÃO DE SISTEMAS**

Este capítulo descreve com detalhes o método para a criação de um SCAPE. O método se utiliza do paradigma de objetos para a modelagem do problema, e da reutilização de produtos de software como técnica de construção. As classes que compõem o modelo são extraídas do processo a ser automatizado, que é organizado e estruturado conforme a arquitetura geral de um sistema de projeto apresentada no capítulo anterior.

#### **3.1 Introdução**

Métodos de engenharia de software são conjuntos de técnicas utilizadas para especificar, projetar e implementar sistemas computacionais. Os problemas de integração de sistemas são tratados pela engenharia de software, mas se diferenciam dos demais métodos de construção de software. Enquanto normalmente o desenvolvimento de software traduz os modelos de projeto em comandos de uma linguagem de programação, na integração de sistemas a tradução é feita em componentes já construídos e prontos para o uso, que são agrupados e operam como um sistema único.

ROZENFELD et al. (1992) descrevem vários requisitos para uma metodologia de integração de sistemas computacionais

para uma empresa, que o método detalhado a seguir, deve atender.

BOOCH (1995) divide os sistemas computacionais em sistemas "de prateleira", que estão prontos para o uso, e sistemas "customizados", que devem ser adaptados às necessidades dos seus clientes. Analisando a aplicabilidade da tecnologia de objetos em ambos os casos, observa que ela visa aumentar os padrões de qualidade do desenvolvimento de software.

A integração de sistemas é considerada como um caso particular de reutilização de software, como observam PUJATTI (1994) e MARTIN (1994). O reaproveitamento dos programas abrevia o ciclo de construção dos sistemas, reduzindo prazos e o custos. Entretanto, a solução final deverá estar limitada a uma combinação de soluções já existentes. Sistemas muito específicos, que não podem ser decompostos em componentes de software existentes, não podem ser atendidos, exclusivamente, por esta técnica.

Como os sistemas para projetos de engenharia são muito semelhantes entre si, a integração de sistemas pode ser aplicada na quase totalidade dos casos. Mesmo quando não houver um programa de computador disponível que atenda aos requisitos do problema, pode-se procurar adaptar algum programa existentes, ou até mesmo desenvolver apenas o programa específico necessário, usando outros métodos de engenharia de software.

Os métodos de engenharia de software existentes não abordam com o detalhe necessário a construção de sistemas pela reutilização de componentes de software. Especialmente na fase de implementação os métodos existentes não são adequados para a construção de sistemas pela integração de produtos de software.

A seguir justifica-se a adoção do paradigma de objetos para a modelagem do SCAPE e apresentam-se as fases do método.

### **3.2 Justificativa da seleção do paradigma de objetos na modelagem de SCAPE**

A adoção de um paradigma permite a simplificação do problema real, e orienta a identificação dos seus aspectos mais relevantes. Dos paradigmas normalmente utilizados pela engenharia de software, o de objetos é o mais adequado para descrever a integração de sistemas, principalmente pela característica de encapsulamento, que ele dispõe.

O paradigma de objetos é o mais recomendado quando se visa a reutilização de software. Este fato decorre do encapsulamento inerente aos objetos, e da possibilidade dada pelos métodos orientados por objetos da separação entre as fases de análise e implementação, permitindo que se trate do problema independentemente de como será construído o sistema. (DEJEZUS, 1995).

Os objetos de um modelo ocultam a implementação dos atributos e das funções que oferecem ao ambiente externo. Encapsulados no componente, estes atributos e funções podem ser reutilizadas por diversos outros sistemas, exatamente como se propõe na reutilização de produtos de software. Os produtos de software também podem ser entendidos como objetos que ocultam a implementação de atributos e funções. Desta forma a implementação pode utilizar o mesmo paradigma usado na análise, simplificando a construção do SCAPE (CONSTANTINE, 1995).

BOOCH (1996) destaca as vantagens de uma abordagem arquitetural no desenvolvimento de um sistema, em oposição ao desenvolvimento centrado em requisitos. Recomenda dividir o desenvolvimento segundo três passos:

- a) especificar o sistemas através de uma série de cenários,
- b) criar uma arquitetura que acomode os padrões encontrados nestes cenários, e
- c) Evoluir a arquitetura na medida em que novos requisitos são descobertos.

O método aqui apresentado segue, aproximadamente, esta proposta, pois parte de uma arquitetura comum para os SCAPE.

A modelagem por objetos também simplifica a manutenção de um sistema computacional, permitindo a inclusão, atualização e exclusão de componentes. Esta característica

é importante quando se deseja alterar parcialmente um sistema existente, como é o caso dos SCAPE.

### **3.3 Técnicas de modelagem utilizadas pelo método**

Na construção do modelo de um SCAPE é importante visualizá-lo sob diversos ângulos, em variados graus de abstração. Assim, utilizam-se diferentes técnicas de modelagem, para decompor o sistema em um conjunto coerente de diagramas.

Neste método são utilizados os seguintes diagramas:

- a) diagrama de domínios, utilizado para delimitar as grandes regiões do sistema, que se relacionam entre si, em um nível alto de abstração. Estabelece uma relação do problema analisado com a arquitetura geral do SCAPE apresentada em 2.3.3.
- b) diagramas de classes, é o diagrama principal, sendo utilizado para descrever os componentes do sistema e suas relações. O diagrama de classes é traduzido no SCAPE durante fase final de implementação do método.
- c) diagramas de transição de estado, são utilizados para descrever a dinâmica de uma classe, identificando os estados que ela pode assumir, quando estimulada por operações internas ou externas. Estes diagramas retratam a evolução de uma classe no tempo, sendo especialmente úteis para modelar controles que agem sobre as classes.

d) diagramas de eventos, um dos diagramas mais utilizados para descrever cenários, representando os processos que ocorrem em um sistema. Permitem analisar as mensagens trocadas entre as classes e as suas reações.

e) diagramas de fluxo de dados, descrevem como as operações transformam as classes. Representam as transformações da informação que ocorrem internamente em cada operação da classe. São raramente utilizados na modelagem de SCAPE, uma vez que as operações neste caso já estão implementadas nos produtos de software e não precisam ser detalhadas.

Existem diversas notações para se construir estes diagramas, que se relacionam aos diferentes métodos de análise e projeto baseado em objetos disponíveis. Pode-se encontrar uma análise comparativa entre estes métodos em HIRAMA (1995) e EMBLEY et al. (1995).

Utiliza-se aqui a notação apresentada pelo método OMT (Object Modeling Technique), desenvolvido por RUMBAUGH et al. (1993). Esta seleção se deve à flexibilidade oferecida por este método para modelar sistemas de um modo geral, e a disponibilidade de uma ferramenta CASE, (MICROGOLD, 1996) para dar apoio à construção dos diagramas e à aplicação da notação.

A notação original de RUMBAUGH et al. (1993) sofrerá pequenas adaptações para considerar as particularidades da

integração de SCAPE. Uma das adaptações é a utilização dos conceitos de modelagem de domínios desenvolvidos por SHLAER, MELLOR (1993).

O anexo C descreve a notação empregada na construção dos diagramas para o método de integração de SCAPE.

### **3.4 Fases do método de integração do SCAPE**

O método de integração de SCAPE segue os princípios básicos de construção dos métodos baseados em objetos. Parte-se de uma estrutura básica, estável, sobre a qual é construída um modelo. Aumenta-se o nível de detalhes do modelo até o ponto de se poder iniciar a implementação.

O método se divide em fases, que identificam estágios na evolução da solução. Cada fase possui objetivos próprios, um conjunto de atividades a serem cumpridas e produz um conjunto de documentos que registram as decisões tomadas.

A granularidade do modelo muda em cada fase. Nas primeiras fases os modelos são bastante abstratos, pois objetiva-se apenas identificar os componentes principais do problema e suas relações. O detalhamento aumenta nas fases seguintes, até conseguir-se identificar nos objetos as funções comuns aos SCAPE, definidas na sua arquitetura geral. Neste ponto da modelagem é possível se iniciar a implementação.

As fases deste método são:

- a) Partição de domínios,

b)Análise interna dos domínios,

c)Projeto do SCAPE e

d)Implementação do SCAPE

A apresentação de cada fase é acompanhada de exemplos genéricos, baseados na própria arquitetura geral de um sistema de projeto (2.3.3).

### **3.5 Fase de Particionamento de Domínios**

Os domínios delimitam regiões do problema com características comuns, e que possuem uma relação cliente-servidor entre si. Cada domínio representa uma parcela do problema, relativamente independente das demais, e que pode ser analisada separadamente. A interligação dos domínios se dá através de pontes, que servem como vias para comunicação e interação.

A vantagem em se iniciar o desenvolvimento pela identificação e particionamento do sistema, é estabelecer uma relação do problema analisado com a arquitetura geral de um SCAPE, apresentada no capítulo 2. Dividindo o sistema em unidades menores, fica mais fácil estudá-lo, e permite que alguns componentes sejam analisados e projetados simultaneamente.

O particionamento dos domínios nos sistemas de apoio à engenharia, deve estar de acordo com o modelo geral do

processo de projeto. Assim, os principais domínios de um sistema de projeto são:

- domínio do produto,
- domínio do gerenciamento,
- domínio do desenvolvimento,
- domínio da infra-estrutura e
- domínio da implementação.

como mostra a figura abaixo:

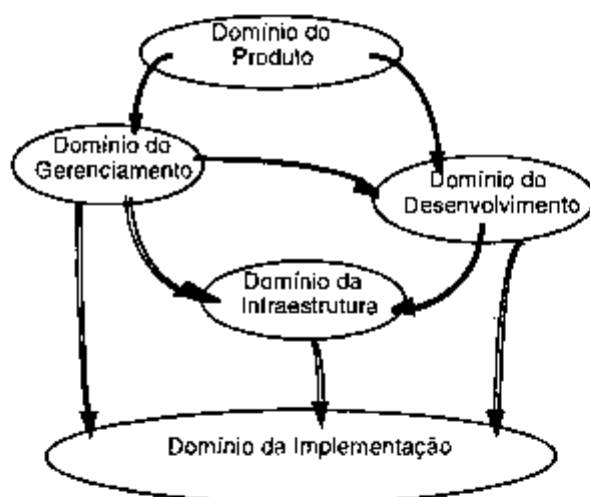


Figura 8 - Diagrama de domínios de um SCAPE

O domínio do produto estabelece os requisitos do produto a ser desenvolvido, e que devem ser atendidos pelos domínios do gerenciamento, traçando os planos de trabalho e pelo domínio de desenvolvimento, definindo as características técnicas do produto. Apoiando domínios superiores deve existir uma infra-estrutura comum de comunicação e de serviços. O domínio do gerenciamento, da infra-estrutura e do desenvolvimento são construídos no domínio da

implementação, que serve a eles todos com produtos de software.

### **3.5.1 Identificação e Particionamento de Domínios**

A procura pelos domínios nos ambientes de projeto parte de uma observação detalhada da organização analisada. Normalmente, não deve haver dificuldade em se identificar os cinco domínios em um problema real, mas pode ser conveniente, para melhor representação do problema, uma partição de domínios diferente.

Pode-se encontrar outros domínios em diferentes linhas de produtos projetados, na estrutura organizacional da empresa, no seu método de trabalho ou em eventuais sistemas computacionais já existentes.

A existência de mais do que uma linha de produtos, que disputam recursos de projeto da organização, pode caracterizar a necessidade de mais de um domínio de produto. Deve-se observar principalmente os diferentes requisitos que cada produto estabelece com os domínios servidores.

Uma organização de projeto pode indicar, pela composição das suas unidades, a existência de subsistemas de projeto independentes. É comum se agrupar em unidades isoladas grupos de projetistas que realizam tarefas semelhantes, e por isso compartilham técnicas e ferramentas de trabalho. Este fato pode ser um indicativo da existência de

subsistemas independentes de projeto, que podem ser modelados como domínios independentes.

O método de trabalho utilizado pela organização pode definir diferentes domínios, ou diferentes relações de serviços entre os domínios. Por exemplo, a existência de etapas de análise, testes ou prototipação no processo de desenvolvimento do produto, pode ser ressaltada, isolando-as em domínios próprios.

A existência de sistemas computacionais já implantados na organização, que se relacionam com o sistema de projeto, mas que não fazem parte do escopo da modelagem, podem ser isolados em um domínio. Pode-se com o diagrama de domínios definir as áreas de atuação do engenheiro de software, isolando-a de outras que não sofrem alterações.

### **3.5.2 Questionário de auxílio na análise de um SCAPE**

A maioria dos casos de automação de projetos é realizada sobre um ambiente de projeto existente. Nestes casos, procura-se melhorar características de desempenho e qualidade do SCAPE atual. Deve-se aproveitar a existência do processo para avaliar quanto ele se aproxima do sistema de projeto ideal, e em qual estágio de evolução no uso da tecnologia se encontra. Esta avaliação indica a tecnologia que deve ser incorporada ao SCAPE atual para torná-lo mais próximo da arquitetura de um SCAPE ideal.

Para um diagnóstico rápido do SCAPE existente, criou-se um questionário para ser submetido aos responsáveis pelo processo. As respostas indicam a existência, ou não, de automação nas diversas funções que um SCAPE deve desempenhar. O anexo B apresenta este questionário, que foi criado com base no trabalho de CARTER; BAKER (1991), adaptado para a arquitetura geral de um sistema de projeto de 2.3.3.

Transformado em um programa de computador, para facilitar a sua utilização, o questionário estimula o responsável pelo sistema a tecer comentários que são úteis durante toda a modelagem do problema.

### **3.5.3 Produtos da fase de partição dos domínios**

Como produtos finais desta fase do método tem-se:

- a) O diagrama de domínios acompanhado de um texto explicativo do escopo de cada domínio, e das relações existentes entre eles.
- b) Um relatório com as respostas do questionário, e uma avaliação preliminar do sistema (opcional).

### **3.6 Fase de análise dos domínios**

A fase de análise dos domínios objetiva registrar o entendimento, por parte do engenheiro de software, do processo que está sendo automatizado e dos requisitos que o sistema deve atender. Procura-se identificar os componentes

do ambiente de engenharia, seus atributos, funções e as relações existentes entre eles, representando-os em diagramas de classes.

Os diagramas são construídos conforme a metodologia OMT (RUMBAUGH et al., 1993), cuja notação está resumida no anexo C. É possível construir-se um diagrama de classes para cada domínio, mas, preferencialmente, deve-se sintetizá-los em um único diagrama, para evidenciar as relações do conjunto.

A análise é feita em um alto nível de abstração, priorizando o entendimento do problema e a clareza da representação nos diagramas. Não devem ser consideradas, neste momento, peculiaridades da implementação pretendida.

### **3.6.1 Caracterização das classes e relacionamentos**

As classes são elementos do problema que possuem uma identidade, armazenam informação e executam operações sobre estas informações. As classes se relacionam entre si, compartilhando dados ou trocando mensagens. São candidatas a serem classes de um modelo os personagens, componentes, documentos importantes e papéis de destaque em cada domínio.

Os diagramas de transição de estado, diagramas de eventos e diagramas de fluxo de dados são diagramas auxiliares na construção dos diagramas de classes. O resultado destes

diagramas serve para aumentar o grau de entendimento do problema, detalhando o diagrama de classes.

Considerando a arquitetura geral de um SCAPE, podemos identificar algumas das suas classes mais comuns, que participam da maioria dos casos. Elas são apresentadas na figura abaixo, e são utilizados como orientação para a identificação de outras classes.

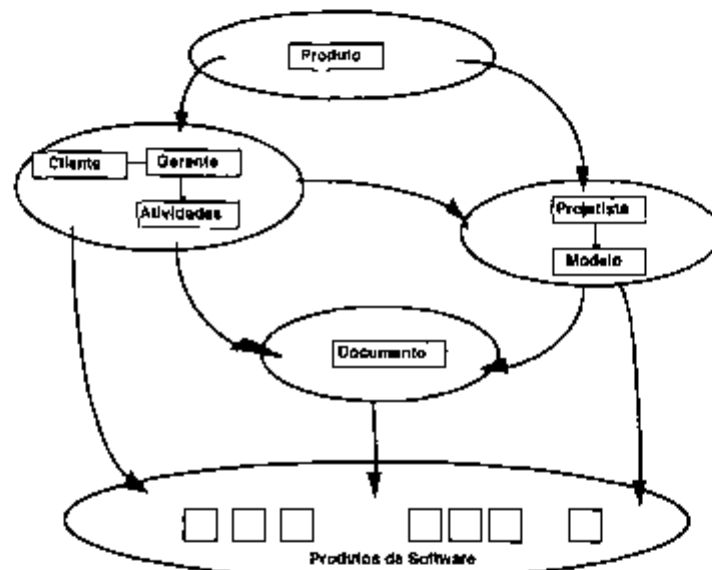
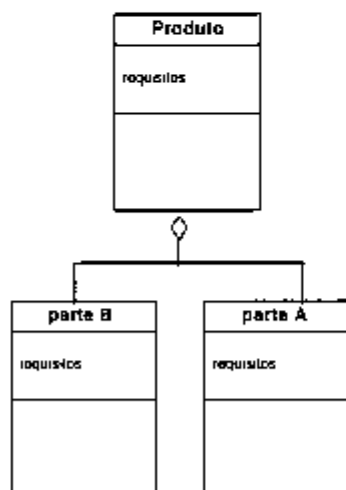


Figura 9 - Principais classes de um SCAPE

No domínio do produto a classe produto é o personagem principal. A ele cabe definir os requisitos que devem ser atendidos pelo SCAPE. Deve-se especializar o produto em suas partes, procurando distribuir seus requisitos entre as especialidades envolvidas no projeto. Esta classe não é implementada, uma vez que não é parte integrante do sistema computacional, sendo utilizada apenas para representar o produto final propriamente dito. Cada parte identificada do produto por receber um subconjunto dos requisitos, e

servirá de tema para o trabalho das equipes de projeto nos domínios servidores. A figura abaixo exemplifica um modelo de objetos do domínio do produto.



*Figura 10 - Diagrama de classes típico do domínio do produto.*

No domínio do gerenciamento procuram-se personagens e elementos usados para o planejamento e controle do andamento do projeto. O personagem principal é o gerente, ou a unidade organizacional que traça as metas de gerenciamento do projeto, como mostra a figura 11.

Contratos, Ordens de Serviço, Planos de Trabalho, Cronogramas estão entre os possíveis elementos que podem compor este domínio. O controle das atividades de projeto pode, por exemplo, ser representado por um diagrama de transição de estado. Os diagramas de transição de estado são usados para descrever os estado que um objeto pode tomar.

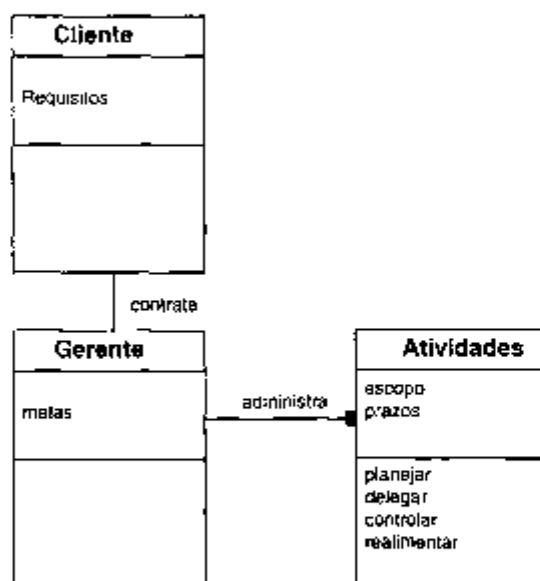


Figura 11 - Diagrama de classes típico do domínio de gerenciamento

Analisando os estados que o objeto atividade pode assumir, e as operações que realizam a alteração destes estados, chegamos, por exemplo, na figura abaixo:

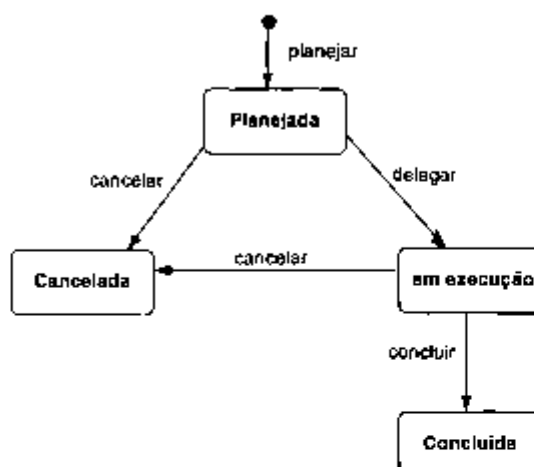


Figura 12 - Exemplo do diagrama de transição de estado de uma atividade

O domínio do desenvolvimento é composto pela classe projetista e pela classe modelos, que representam os modelos matemáticos e gráficos desenvolvidos sobre o

produto. Podem também ser candidatos a objetos as técnicas empregadas e outros produtos do projeto, como protótipos de ensaio e maquetes.

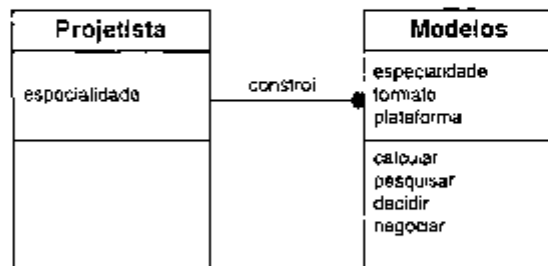


Figura 13 - Diagrama de classes típico do domínio de desenvolvimento.

Os diagramas de estado podem ser usados para descrever a interação que ocorre entre as classes projetistas, atividades e gerentes. A figura abaixo mostra que os eventos estimulam os objetos e provocam outros eventos, estabelecendo uma seqüência de operações.

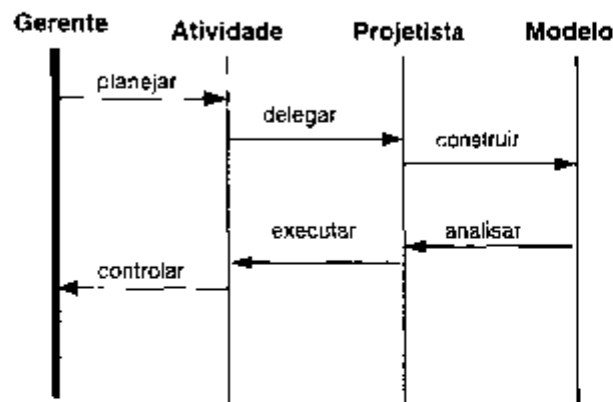


Figura 14 - Exemplo do diagrama de eventos de um processo

O domínio da infra-estrutura serve aos domínios do gerenciamento e de desenvolvimento. Neste domínio a unidade da informação é o documento de projeto, que se constitui no seu principal componente. A infra-estrutura pode estar

associada também aos serviços de entrada e saída de informação em diversos meios, a elementos de comunicação de dados e aos meios de armazenagem de informação.

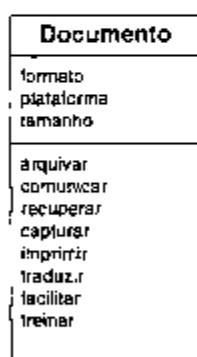


Figura 15 - Exemplo de classe típica do domínio de infraestrutura.

Uma vez conhecidas as classes nos diversos domínios deve-se determinar os atributos e as operações ligadas a cada um deles. Os atributos são informações estáticas pertencentes ao objeto, compondo a sua estrutura de dados. As operações são ações executadas pelo objeto que modificam os atributos. Os atributos e as operações são encapsulados nos objetos do modelo.

As associações entre os classes são identificadas para descrever os vínculos existentes entre elas, as hierarquias e as especializações de cada uma. Estas associações são úteis para se estudar o acoplamento entre as classes, e para compartilhar atributos e operações.

### 3.6.2 Produtos da fase de análise

A análise é uma fase interativa de modelagem, que ocorre em um grau alto de abstração. O término da análise ocorre, a

critério do analista, quando houver um razoável entendimento do processo que está sendo automatizado. Até obter este entendimento deve-se repetir na análise na busca de maior compreensão.

Como resultado desta fase os seguintes documentos são produzidos:

- a) diagramas de classes dos domínios,
- b) diagramas de transição de estados e diagramas de eventos dos objetos que representam situações reais (opcional),
- c) diagramas de fluxo de dados para esclarecer as operações mais complexas (opcional), e
- d) glossário de termos explicando o significado dos nomes dos objetos, atributos e operações (opcional).

### **3.7 Fase de projeto do SCAPE**

A fase de projeto de sistemas visa orientar os diagramas produzidos na análise na direção da implementação. As classes são detalhadas e tem seus atributos e operações classificados com vistas à integração de produtos que as constituirão.

#### **3.7.1 Definição da plataforma de processamento**

A fase de projeto do SCAPE tem início com a escolha da plataforma de processamento e da distribuição das classes entre os processadores. São estabelecidos padrões para

serem utilizados pelos equipamentos e sistema operacional. Estas definições servem para orientar a escolha da tecnologia de integração a ser empregada.

A arquitetura distribuída cliente-servidor, é a mais adequada para os sistemas de apoio a projetos de engenharia. Neste caso as classes são distribuídas entre os processadores que se integram em uma rede de comunicação de dados. Esta arquitetura permite a especialização dos processadores nas tarefas de servidores ou clientes.

O modelo dos processadores dependerá das funções que as classes processadas ali irão desempenhar. A quantidade de processadores e sua configuração dependerá do desempenho desejado para o sistema, e das possibilidades de simultaneidade nas operações, e só deverão ser definidos após o projeto de software do SCAPE. O hardware, neste tipo de sistema, é configurado dentro de uma família de processadores após ter sido definido o software a ser processador. A configuração do hardware deve garantir a performance adequada ao sistema, sem aumentar o seu custo excessivamente.

Deve-se também estabelecer, juntamente com a arquitetura do hardware, o sistema operacional e o padrão das interfaces dos operadores com o SCAPE. Estas definições devem, preferencialmente, ser selecionadas entre os padrões de sistemas abertos, aumentando a disponibilidade de produtos

de software para atender à implementação, além de favorecer os requisitos de portabilidade e flexibilidade do SCAPE.

### **3.7.2 Detalhamento do diagrama de classes**

Definida a plataforma de processamento, deve-se analisar o seu efeito sobre os modelos resultantes da análise. Os formatos de arquivos, sistema operacional e tipo de processadores devem ser registrados nos objetos de cada domínio, como novos atributos das classes.

Classes decorrentes de necessidades de implementação, como gerenciadores de base de dados, tradutores de formato, armazenamentos intermediários devem ser incluídos nos diagramas neste momento.

O momento é adequado também para se aumentar o grau de detalhamento dos modelos, simplificando as heranças, duplicando classes, redistribuindo atributos ou funções com objetivo de facilitar a implementação.

Pode-se, como mostra a figura 16 , ao passar da fase de análise para a fase de projeto, decompor classes complexas em unidades menores. Esta decomposição é acompanhada pela distribuição de funções entre os objetos resultantes. Deve-se procurar distribuir operações que facilitem a seleção de um produto, agrupando funções e atributos que tenham alguma afinidade entre si.

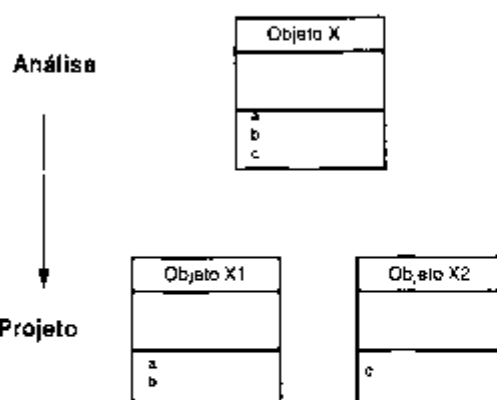


Figura 16 - Exemplo de operação nos diagramas para facilitar a tradução

As operações de cada classe devem ser classificadas nesta fase. A classificação é feita escolhendo-se para cada uma das operações de uma classe um tipo, relativo à família de funções definidas pela arquitetura geral do SCAPE. A figura abaixo mostra um exemplo da classificação das funções de um objeto.

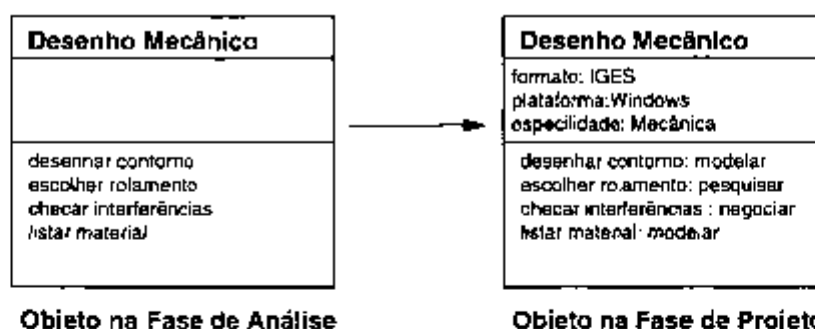


Figura 17 - Exemplo de detalhamento de classe

### 3.7.3 Produto da fase de projeto de sistemas

O produto final desta fase do desenvolvimento dos sistemas são os diagramas de classes detalhados. O nível de detalhe é definido pelo projetista e deve ser suficiente para possibilitar a seleção de produtos para a implementação.

### **3.8 Fase de implementação do SCAPE**

A implementação traduz o modelo de classes gerado no projeto, em programas de computador que formam um SCAPE. Pesquisa-se no conjunto de programas de computador disponíveis, aqueles que atendem aos requisitos técnicos da área de especialização da classe, e que executem as operações previstas, obedecendo aos tipos dos atributos de formato e plataforma adotados.

#### **3.8.1 Seleção das classes aqui implementadas**

Nem todas as classes do diagrama são traduzidas em programas de computador. Não são traduzidas as classes que representam os elementos pertencentes ao processo mas que não serão automatizados, ou que foram criados para facilitar o entendimento do processo, mas que não representam componentes de software do sistema.

Não são implementadas as classes relativas aos produtos ou aos personagens humanos do processo, que não são automatizados como o projetista ou o gerente.

#### **3.8.2 Tradução das classes em produtos de software**

Identificadas as classes que serão implementadas, pode-se consultar o banco de produtos na tentativa de localizar produtos de software que atendam aos seus requisitos. Numa situação ideal, onde existe um único produto de software que atende completamente as necessidades de uma classe,

pode-se dizer que o produto é uma instânciação daquela classe, e incluí-lo no domínio da implementação. A notação utilizada será:

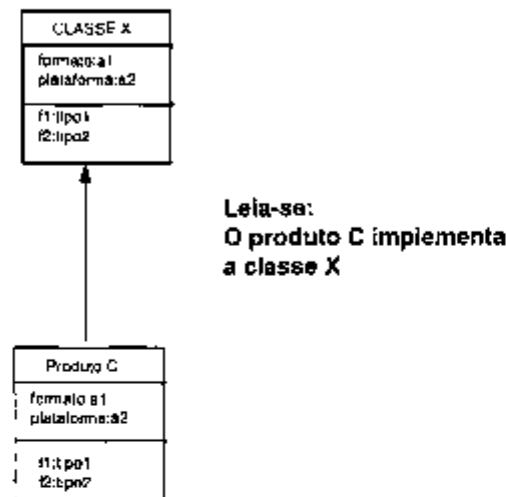


Figura 18 - Exemplo de instânciação de produtos de software

Entretanto, nem sempre esta relação direta é possível. Podem existir produtos que atendam às funções de várias classes, assim como podem existir classes que necessitem de mais do que um único produto para satisfazer seus requisitos. Neste caso utiliza-se a mesma notação de uma seta do produto para a classe:

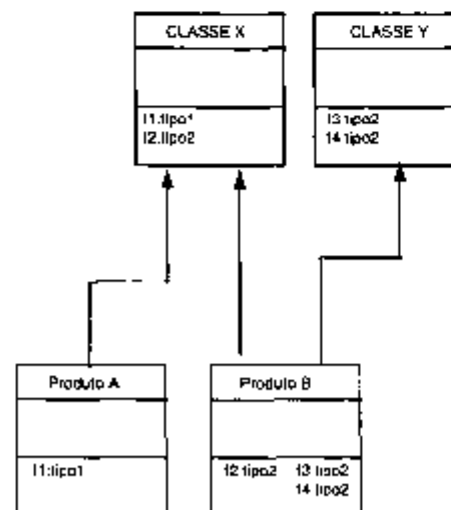


Figura 19 - Relacionamento entre classes e produtos

Como mais do que um único conjunto de produtos de software pode atender as necessidades de um SCAPE, as soluções possíveis não são únicas. A escolha da melhor solução é obtida estabelecendo um critério adicional de seleção. Este critério é definido pelo projetista em conjunto com seu cliente para cada caso. São possíveis critérios: a disponibilidade dos produtos, o custo da implementação ou por exemplo a familiaridade dos projetistas com alguns produtos. Pode-se até realizar testes comparativos entre os produtos candidatos para realizar a seleção. Como fogem do caso geral, o estudo dos critérios de seleção não faz parte deste trabalho, se constituindo em uma linha de pesquisa futura.

### **3.8.3 Produtos da fase de implementação**

A etapa de implantação produz:

- a) Relação dos produtos de software que compõem o SCAPE, e
- b) Especificação técnica de cada produto.

### 3.9 Resumo do método

A tabela a seguir ilustra os passos do método:

<b>Fase de Particionamento de Domínios</b>	
<ul style="list-style-type: none"> <li>• Identificação e Particionamento de Domínios</li> <li>• Aplicação do Questionário de Auxílio à Análise</li> </ul>	
<b>Fase de Análise dos Domínios</b>	
<ul style="list-style-type: none"> <li>• Identificação das classes e Modelagem dos processos</li> </ul>	
<b>Fase de Projeto do SCAPE</b>	
<ul style="list-style-type: none"> <li>• Detalhamento dos diagramas de classes,</li> <li>• Seleção da arquitetura de processamento,</li> <li>• Classificação das operações e atributos</li> </ul>	
<b>Fase de Implementação do SCAPE</b>	
<ul style="list-style-type: none"> <li>• Seleção das classes a serem implementadas,</li> <li>• Seleção os produtos de software para atender à cada classe</li> </ul>	

Tabela 6 - Tabela ilustrativa que resume o método

## **4. EXPERIMENTOS DE LABORATÓRIO E DE APLICAÇÃO DO MÉTODO**

Este capítulo descreve três casos de utilização do método:

- Uma célula de trabalho para o engenheiro projetista,
- Um SCAPE aplicado a projetos de navios, e
- Um SCAPE aplicado a projetos nucleares.

Os dois primeiros visam o entendimento do método, sendo aplicações exclusivamente laboratoriais. O terceiro exemplo trata-se de uma caso real extraído da experiência pessoal do autor na implantação de sistemas de apoio à engenharia nos projetos desenvolvidos no CTMSP da Marinha do Brasil. Inicialmente, apresenta-se o laboratório onde são desenvolvidos estes exemplos.

### **4.1 Descrição do laboratório de experimentos**

O laboratório para a realização dos experimentos é composto pelo conjunto de programas de computador que dá suporte ao projetista de SCAPE. A figura 20 esquematiza a configuração do ambiente computacional utilizado para elaborar os experimentos. Este ambiente reside em um microcomputador com processador INTEL modelo Pentium de 100 MHz com 16 MBytes de memória no sistema operacional Windows 95.

As fases iniciais de análise e projeto do método são apoiadas por um programa CASE, que permite construir, graficamente, os diagramas de eventos e classes dos sistemas estudados. Utilizou-se para isto o programa WITH CLASS 3.0 (MICROGOLD, 1996), que obedece a metodologia OMT (RUMBAUGH et al., 1991) e permite gerar relatórios descritivos que podem ser associados aos diagramas.

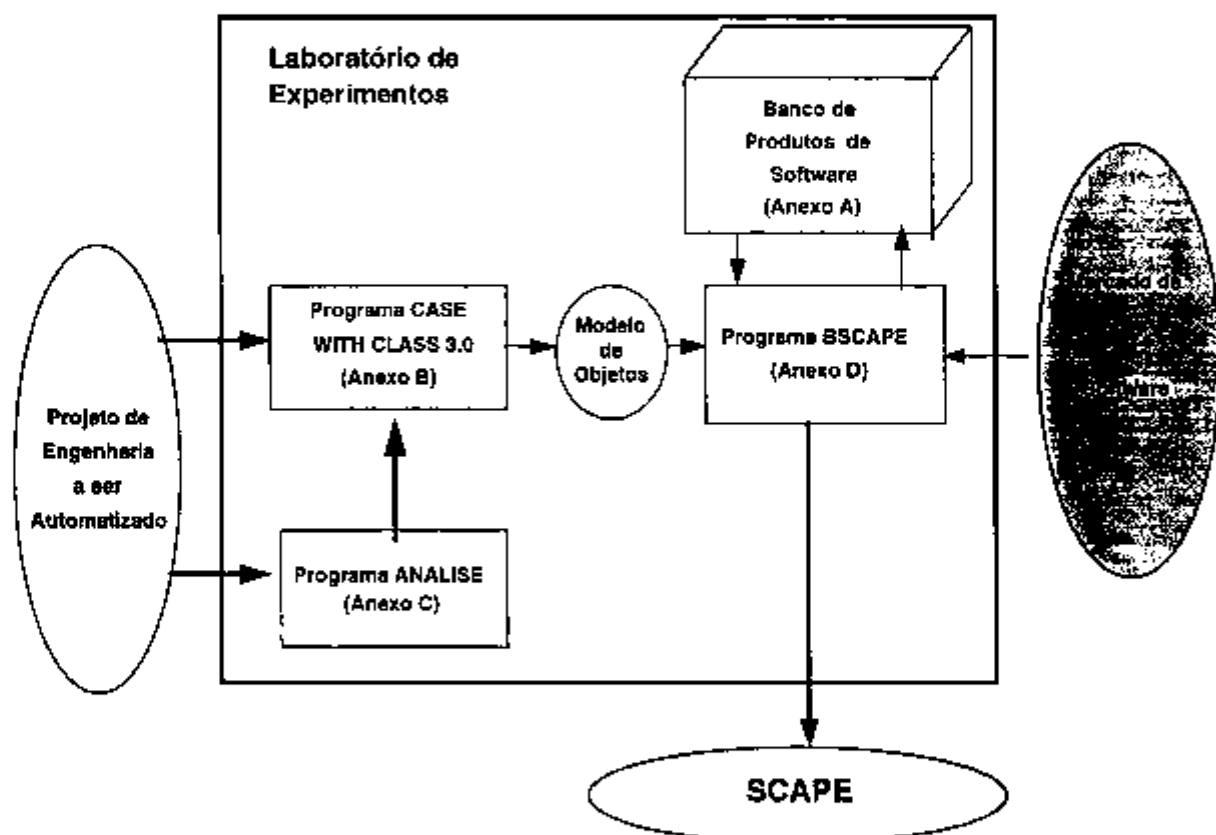


Figura 20- Esquema do Apoio Computacional do Laboratório de Experimentos

A figura seguinte mostra um exemplo da interface do programa WITH CLASS. A sua interface gráfica facilita a criação e modificação dos diagramas do modelo de objetos.

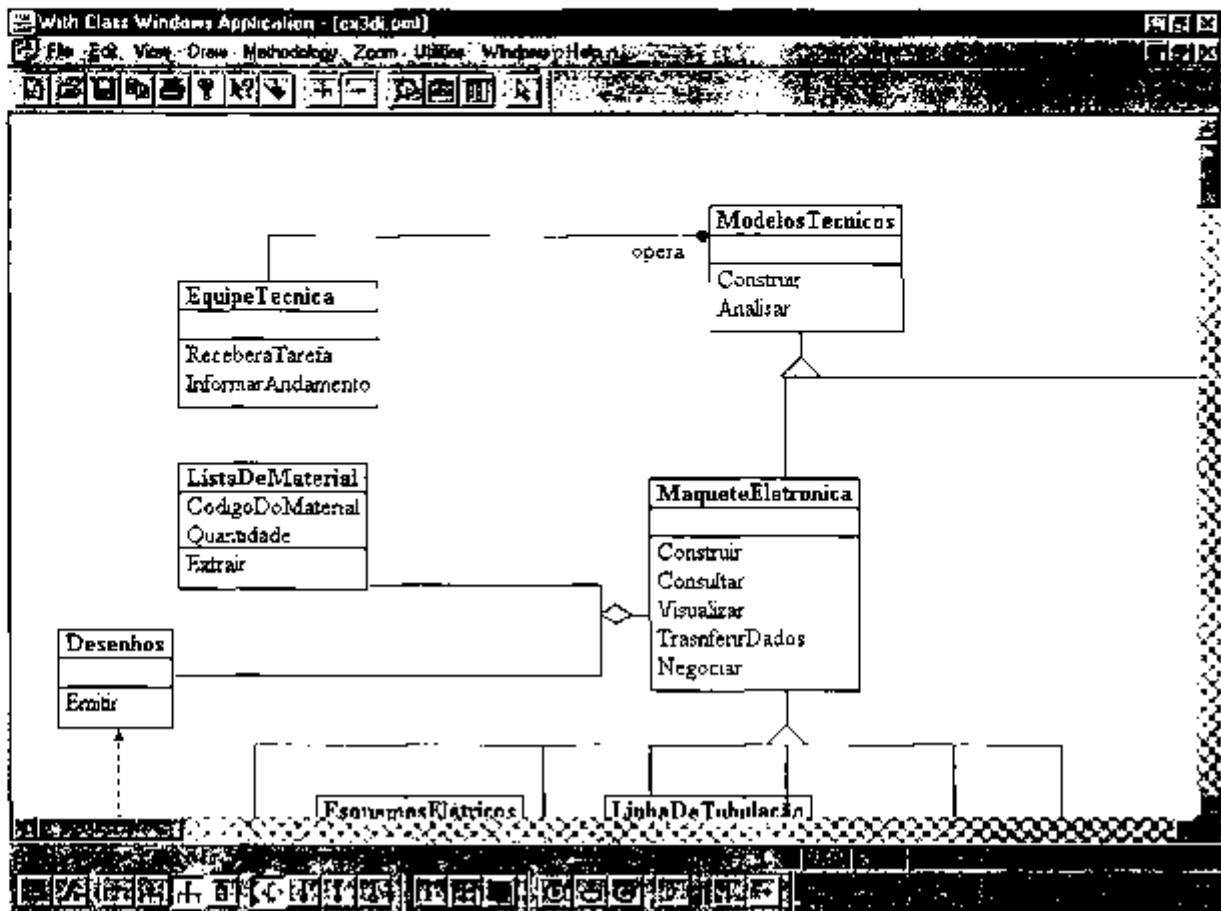


Figura 21 - Exemplo da interface do programa *WITH CLASS* 3.0.

Para facilitar uma análise inicial dos sistemas que apoiam projetos de engenharia desenvolveu-se o programa *ANALISE*. Este programa automatiza o questionário para avaliação de um *SCAPE*, apresentado no anexo C. Através deste programa gera-se um relatório de análise do sistema existente, que serve de base para a sua modelagem.

O Banco de Produtos de Software, especificado no anexo A, foi construído no sistema gerenciador de base de dados Access 2.0 (MICROSOFT, 1992a). Estes dados são atualizados pelo programa *BSCAPE*, apresentado no anexo D, que foi criado na linguagem Visual Basic 4.0. Foi possível integrar

ao Banco de Produtos os dados do ANUÁRIO DE INFORMÁTICA 96/97 (1996), com mais de 2000 endereços de empresas do setor com boa abrangência como fonte de informação sobre produtos e referência de fornecedores.

Durante a realização dos experimentos o banco de produtos encontrava-se com cerca de 250 produtos de software cadastrados. Estes se distribuíam entre as funções padronizadas no capítulo 2, como mostra o gráfico abaixo:

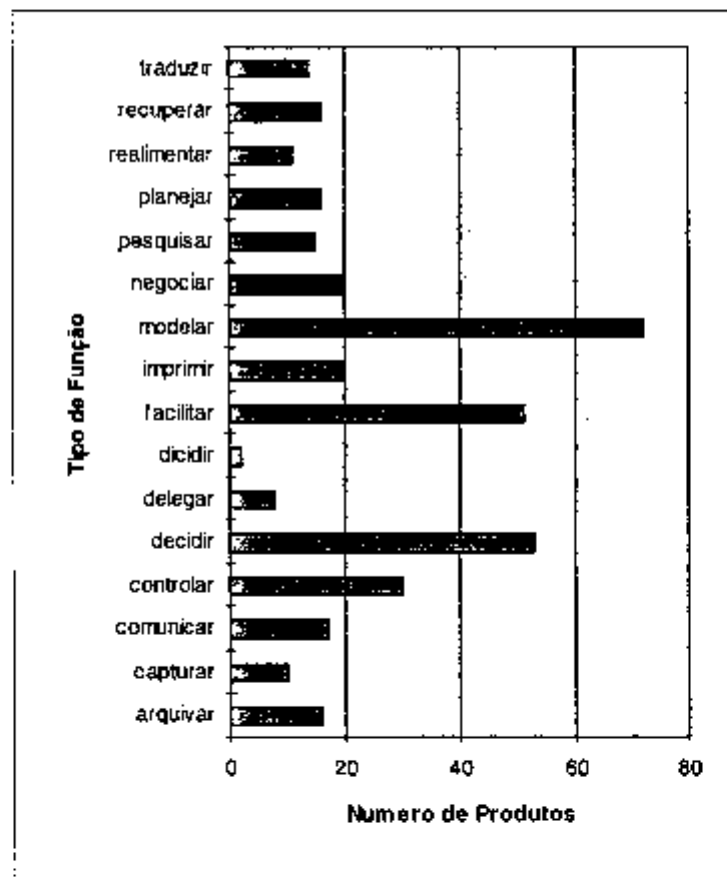


Figura 22 - Distribuição dos produtos cadastrados entre as funções

A fase de implementação do método é apoiada pelo próprio programa BSCAPE, desenvolvido pelo autor, especificamente para apoiar o método. Este programa, descrito no anexo D,

recebe como entrada o diagrama de classes detalhado resultante da fase de projeto. O programa permite ao projetista de SCAPE selecionar os produtos de software que serão utilizados na implementação de cada classe do modelo. A figura abaixo mostra um exemplo da tela do programa BSCAPE durante a seleção de um produto.

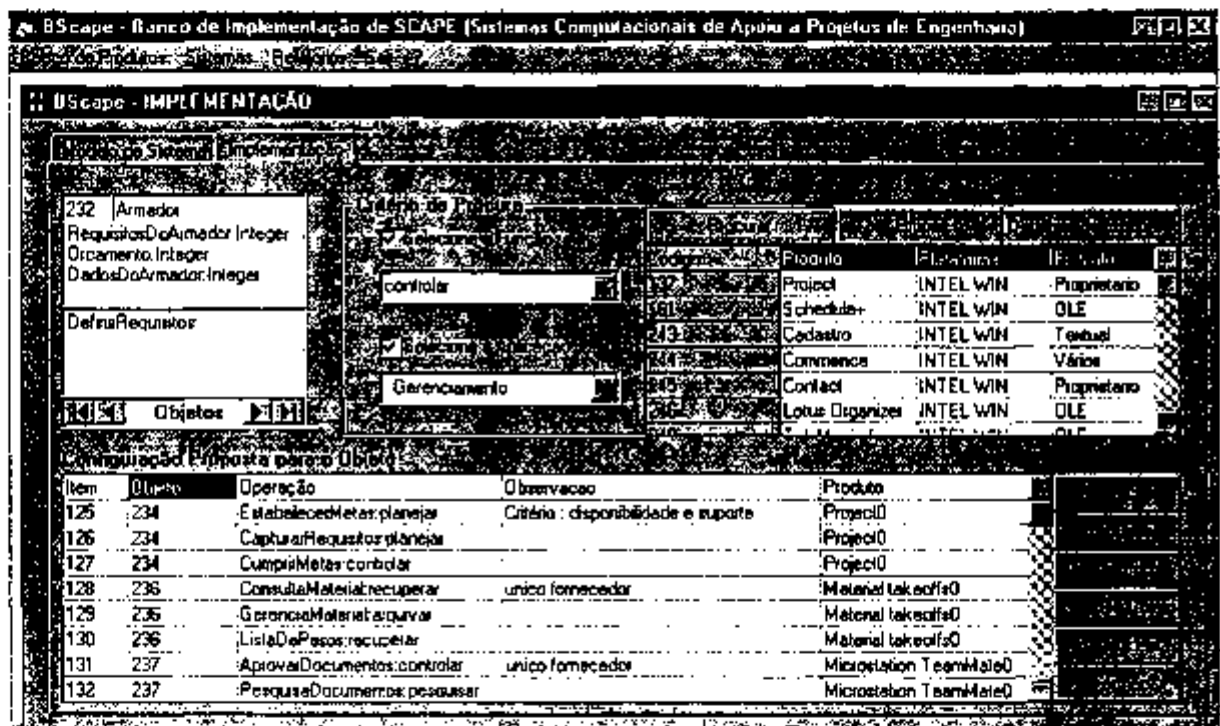


Figura 23 - Amostra da interface do programa BSCAPE

Visando a integração das fases de análise, projeto e implementação, adaptou-se o programa WITH CLASS para gerar arquivos com a descrição dos modelos de SCAPE. Estes arquivos são lidos pelo programa BSCAPE, permitindo uma passagem automática do diagrama da fase de projeto para a fase de implementação. O programa CASE também foi adaptado para gerar relatórios descritivos dos modelos analisados, simplificando a documentação de cada diagrama.

#### **4.2 Caso 1: Célula de trabalho para o engenheiro projetista**

Este exemplo descreve a criação de um sistema computacional de apoio ao engenheiro projetista em seu posto individual de trabalho. Objetiva-se criar uma célula flexível de trabalho, que possa ser interligada a outras, através de uma rede de comunicação de dados, para a formação de uma empresa virtual de projeto. Para assegurar generalidade ao exemplo, não se focalizou o estudo em nenhuma área específica de especialização do projetista, limitando-se a considerar exclusivamente os aspectos comuns a todos os projetistas.

Este projetista hipotético produz somente documentos de projeto. Estes documentos, em formato eletrônico, podem ser desenhos ou relatórios técnicos com textos e figuras. O posto de trabalho recebe pedidos de projeto através de uma rede de comunicação de dados, e por ela também transmite os documentos produzidos.

Este exemplo de caráter exclusivamente laboratorial, aplica os conceitos desenvolvidos pelo método proposto. Como é um exemplo hipotético, não é possível se utilizar o questionário de auxílio à análise descrito no anexo C.

#### 4.2.1 Particionamento dos domínios do caso 1

Aplica-se, inicialmente, ao posto individual de trabalho, a estrutura típica de um projeto de engenharia apresentada em 2.3.3. Baseando-se no particionamento dos domínios de um SCAPE típico, observa-se que como o exemplo não especificou uma área de aplicação, o domínio do produto não exerce grande influência no sistema, e portanto não será considerado. Restam para serem analisados os domínios de Gerenciamento, Desenvolvimento, Infra-estrutura e logicamente o domínio da Implementação. O diagrama de domínios do exemplo é representado pela figura:

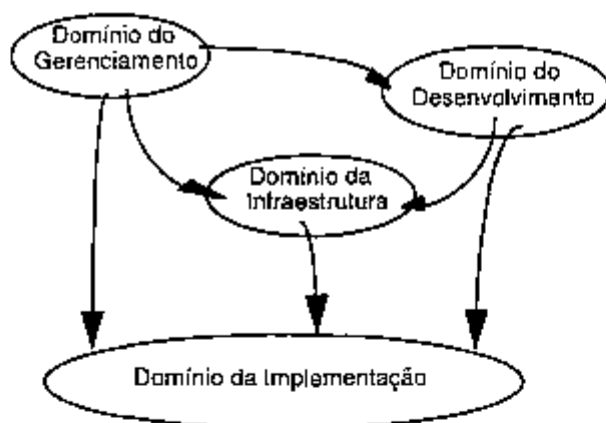


Figura 24 - Diagrama de domínios da célula de trabalho

#### 4.2.2 Análise dos domínios do caso 1

Na fase de Análise busca-se o entendimento dos processos realizados em cada domínio. Os processos são descritos através de diagramas de eventos que, posteriormente, serão integrados em um diagrama de classes.

O domínio do planejamento descreve as etapas de programação que antecedem o desenvolvimento de um projeto. O engenheiro recebe os pedidos de um solicitante, pela rede de comunicação de dados, acompanhados dos documentos técnicos onde encontra informações importantes para o seu trabalho. O projetista programa a solução do problema informando ao solicitante, em uma proposta, o prazo de execução e o custo do projeto. Aguarda-se então que a proposta seja aprovada para iniciar os trabalhos. A estimativa de custos e prazo é decorrente do planejamento das atividades do projetista e do tempo que será consumido em executá-las. Estes cenários são descritos pelas figuras abaixo, na forma de diagramas de eventos.

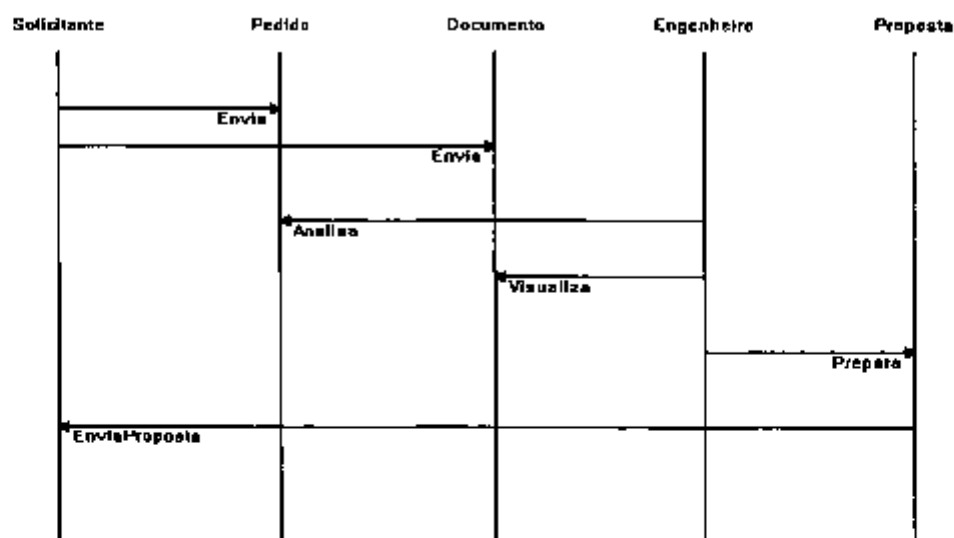


Figura 25 - Diagrama de eventos da solicitação de um Estudo

O desenvolvimento do projeto realizado na célula resume-se na criação, análise e modificação de documentos de projeto. Estes documentos são baseados em documentos com informação do tipo texto ou figura. Cada especialidade de engenharia

utiliza diferentes programas de computador para este trabalho, mas que não serão considerados neste exemplo. Considera-se aqui um documento de projeto genérico, que deverá ser especializado para estender a aplicação desta célula a um projeto de engenharia real.

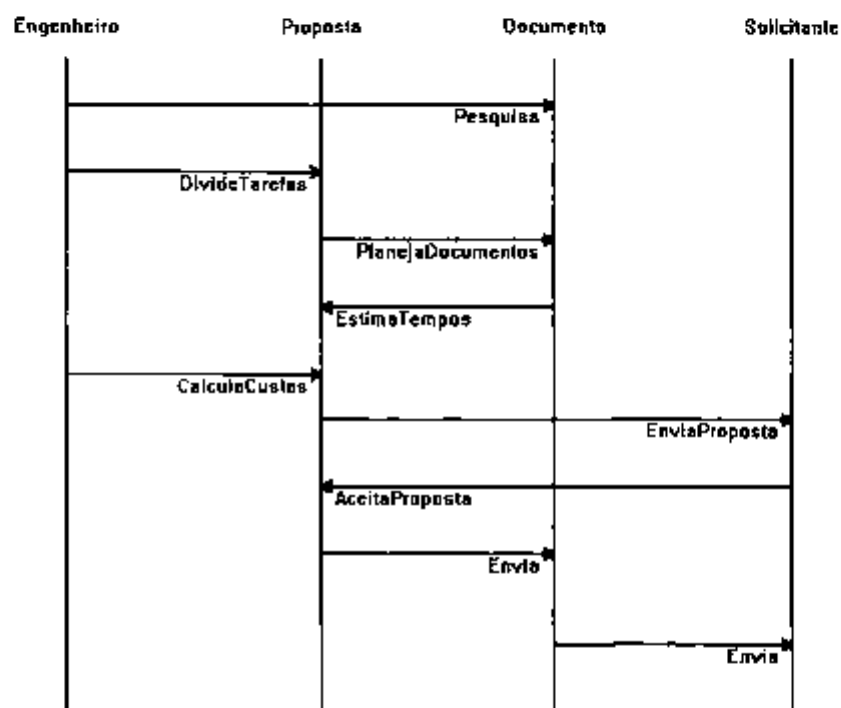


Figura 26 - Diagrama de eventos da preparação de uma proposta

A aceitação da proposta pelo solicitante caracteriza o início dos trabalhos. Segue-se a pesquisa e análise dos documentos recebidos, que servem como fonte para a criação de novos documentos de projeto. Uma vez concluídos, eles são enviados ao solicitante para análise e aprovação. Este processo é representado pela figura que se segue:

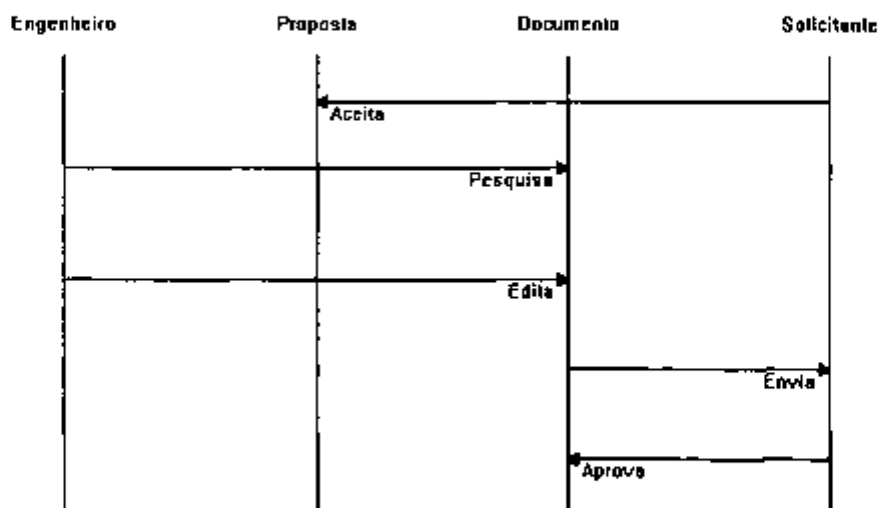


Figura 27 - Diagrama de eventos da edição de um relatório técnico

O domínio da infra-estrutura cuida dos aspectos de armazenamento dos documentos e da relação destes com a rede de comunicação de dados. Neste domínio a informação está encapsulada em um arquivo. O diagrama abaixo mostra algumas das operações que ocorrem com o arquivo, neste domínio.

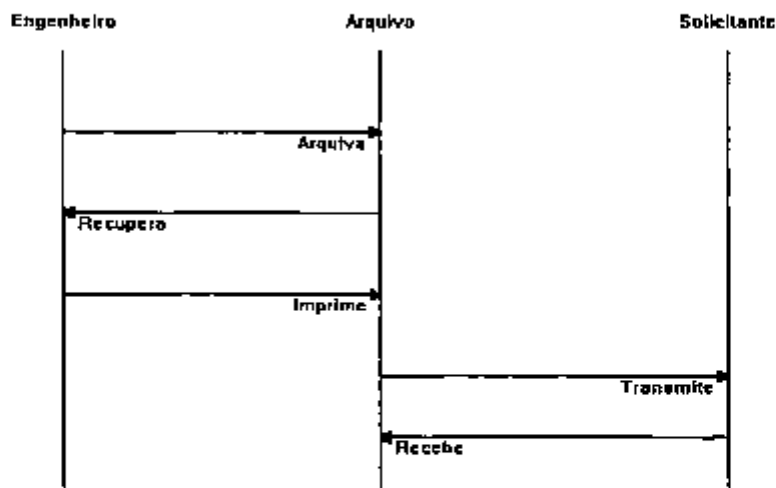


Figura 28 - Diagrama dos eventos no domínio da infra-estrutura

#### 4.2.2.1 Diagrama de classes

O diagrama de classes, na fase de análise, é formado pela integração das diversas visões apresentadas nos diagramas de eventos. A simplicidade deste exemplo permite que um único diagrama de classes represente todo o sistema.

As classes relativas ao projetista e ao solicitante não são representadas, por se tratarem de classes externas ao sistema. Por entre as classes do diagrama são distribuídas as responsabilidades do sistema, identificadas nas operações dos diversos processos. Neste diagrama estão também representados os atributos que identificam e caracterizam cada classe. A figura abaixo apresenta o modelo de objetos resultante da fase de análise do problema.

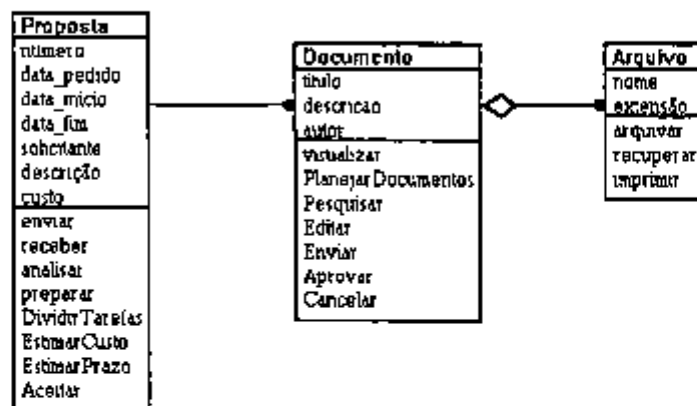


Figura 29 - Diagrama de classes resultante da análise de domínios

#### 4.2.3 Projeto do SCAPE para o caso 1

Na fase de projeto de sistemas o diagrama de classes da análise é enriquecido com detalhes conduzindo-o à implementação. Incorpora-se elementos da arquitetura na qual será processado o sistema, definindo-se, entre outros, a plataforma de hardware e o sistema operacional que serão utilizados.

Para este experimento, o posto de trabalho será processado na plataforma mais comum aos escritórios de engenharia: Microcomputadores da linha PC com processadores INTEL, e sistema operacional Windows.

Estas definições entram como atributos constantes nas classes. Aumentado-se o grau de detalhe do modelo especializando a classe arquivo nos seus diversos tipos. As operações são analisadas e redistribuídas entre as classes visando reduzir o acoplamento entre as classes. As operações de cada classe são estudadas e classificadas segundo os critérios definidos pela tabela 2.2. Esta classificação ajudará na localização dos produtos do banco que podem implementar a classe. A seguir temos o diagrama de classes resultante:

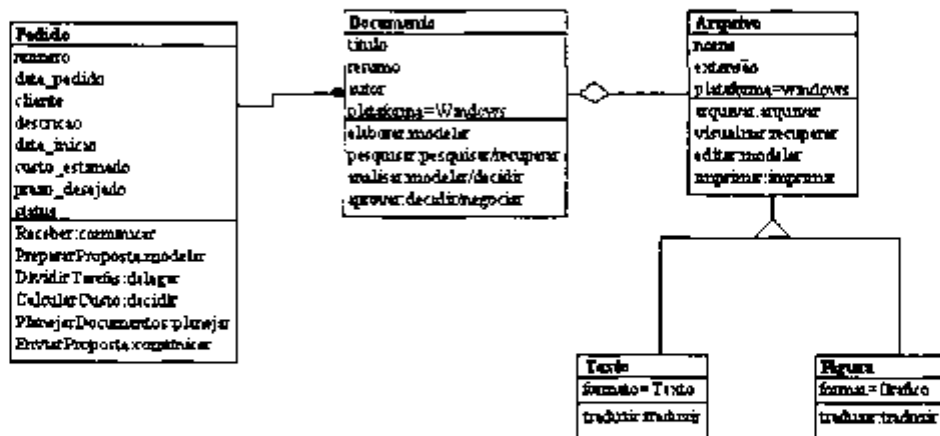


Figura 30 - Diagrama de classes resultante do projeto.

#### 4.2.4 Implementação do SCAPE no caso 1

O objetivo da implementação é identificar um conjunto de produtos de software que possam executar as operações propostas para cada classe. O diagrama de classes é transferido do programa CASE para o programa BSCAPE. Cria-se um arquivo do diagrama, em formato especial, com a descrição dos objetos, que pode ser lido pelo programa BSCAPE.

O programa BSCAPE permite a criação de uma lista de produtos de software que atendam às diversas operações de cada objeto. Esta lista de produtos candidatos contém um resumo do produto, suas funções principais, os dados do fabricante e de possíveis fornecedores.

Os produtos candidatos são então avaliados individualmente pelo próprio projetista. O critério de avaliação deve levar em conta não apenas aspectos técnicos, mas também aspectos comerciais, disponibilidade do produto, facilidade de uso

entre outros. Se o conjunto de informações que o banco de dados dispõe não for suficiente para a seleção, deve-se realizar uma consulta aos fornecedores para maiores informações. A avaliação dos produtos deve ser realizada em conjunto para todos as classes, uma vez que um produto pode atender a operações de mais do que uma só classe. Os requisitos da arquitetura de processamento escolhida elimina uma parcela dos produtos candidatos.

Selecionando-se um conjunto de produtos que executa todas as operações necessárias, chega-se a uma configuração para o SCAPE. O programa BSCAPE pode gerar os relatórios e as especificações finais do sistema para a sua aquisição. No exemplo a configuração proposta para o SCAPE de um engenheiro projetista é apresentada na tabela abaixo.

Objeto	Operação: tipo	Produto
Pedido	Receber:comunicar EnviarProposta:comunicar	COMMSUITE
Pedido	PlanejarDocumentos:planejar PrepararProposta:modelar	Project
Pedido	CalcularCusto:decidir	EXCEL
Documento	aprovar:decidir	InVision V. 3.0
Documento	pesquisar:pesquisar	Falcom/DMS 5.1
Arquivo	visualizar:recuperar	Myriad
Texto	traduzir:traduzir	Myriad
Figura	traduzir:traduzir	Myriad

Tabela 7 - Levantamento de produtos para implementação do Caso 1

Onde os programas escolhidos são:

Produto	Descrição
COMMSUITE	Programa de comunicação de dados, transferência de arquivos e comunicação com a Internet. Inclui procedimentos de segurança.
Project	Gerenciador de projetos com uma série de assistentes para facilitar a montagem e supervisão de qualquer projeto. Auxilia a colocar em prática os planos, comunicá-los a outras pessoas e administrar as modificações
EXCEL	Planilha eletrônica de cálculos. Recursos gráficos de apresentação.
InVision V. 3.0	Software para videoconferencia, Suporte a rede Ethernet, com whiteboard para desenho. O mais simples de todos os sistemas, resolução de 160 x 120, possui funções de privacidade.
Falcom/DMS 5.1	Gerenciador de documentos para usuários Microstation Organiza os documentos em diretórios (6 níveis), possui um controle de tempo gasto em cada desenho, para efeito de contabilidade
Myriad	Visualizador de arquivos em diversos formatos. Anotação (redlining) de desenhos e documentos. Revisão de documentos. Impressão e Plotagem de documentos eletrônicos.

Tabela 8 - Descrição dos produtos selecionados

Pode-se, opcionalmente, voltar ao modelo de objetos e incluir os objetos referentes aos produtos de software, associando-se aos objetos do modelo, como mostra o diagrama:

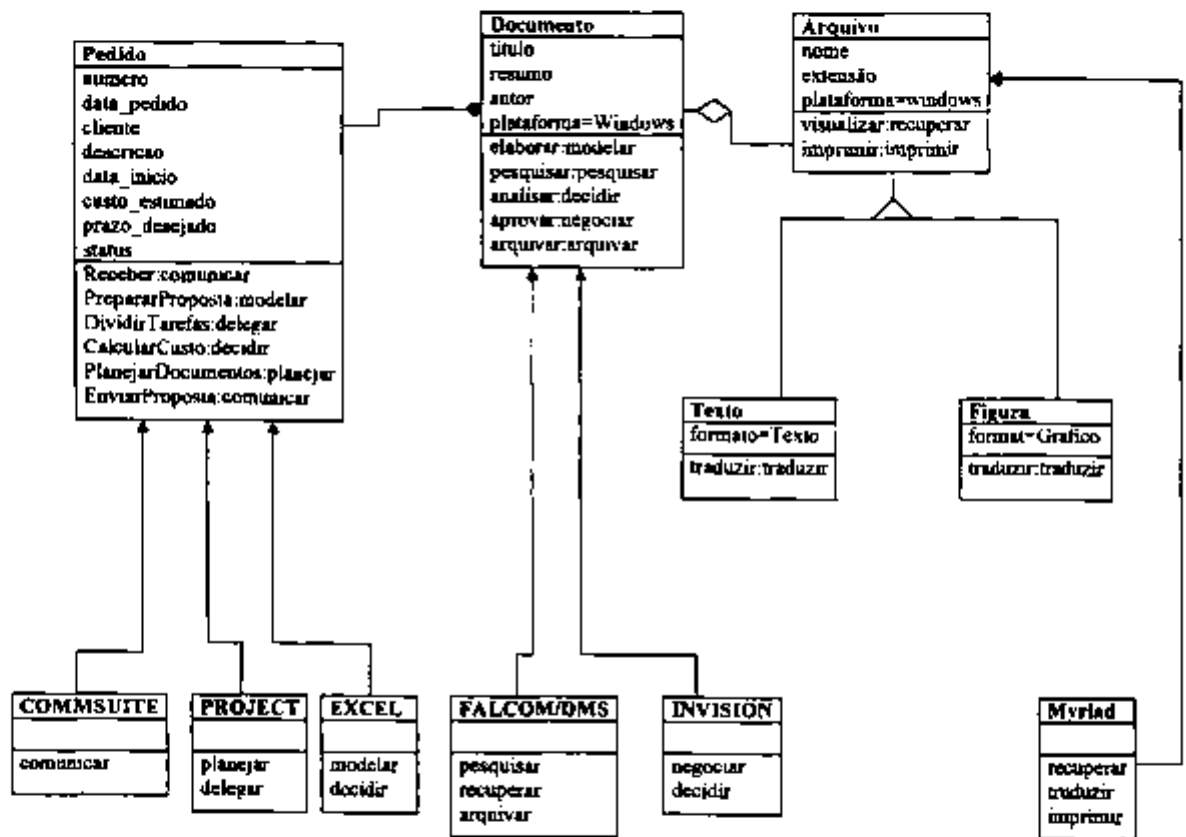


Figura 31 - Diagrama de Classes final do sistema

### **4.3 Caso 2: Sistema Computacional de Apoio ao Projeto de Navios**

Este exemplo descreve a criação de um sistema computacional de apoio ao projeto de navios, para ser utilizado por um escritório de projetos típico em um estaleiro. O objetivo do estudo é criar um SCAPE onde todas as atividades de projeto sejam realizadas através do computador, baseando-se exclusivamente, em produtos de software disponíveis no mercado de informática.

#### **4.3.1 Particionamento dos domínios do caso 2**

A divisão dos domínios, é feita de acordo com o particionamento de um SCAPE apresentado em 3.5. Neste exemplo, ao contrário do caso anterior, a complexidade do produto (navio) exerce uma grande influência sobre todo o sistema. O número de disciplinas envolvidas no projeto de uma embarcação exige uma grande especialização do domínio do desenvolvimento, assim como impõe condições especiais para seu gerenciamento.

#### **4.3.2 Análise dos domínios do caso 2**

Nesta fase do método são criados diagramas para o entendimento dos processos ocorridos nos diversos domínios. Após a localização das classes de cada domínio, estudam-se os diferentes processos que ocorrem entre elas traçando-se diagramas de eventos para os processos principais. Os diagramas de eventos ajudam a definir as operações das classes do SCAPE.

#### 4.3.2.1 Análise do domínio do produto: navio

Neste domínio encontram-se os componentes principais do produto e seus requisitos. As relações encontradas entre as classes deste domínio são relações de agregação, pois representam a decomposição de um todo (navio) em suas partes. A figura abaixo mostra uma primeira aproximação para o diagrama de classes deste domínio. Os atributos de cada classe são associados aos requisitos do produto. O produto é decomposto até o ponto em se identificam os modelos e as atividades necessárias para o seu projeto.

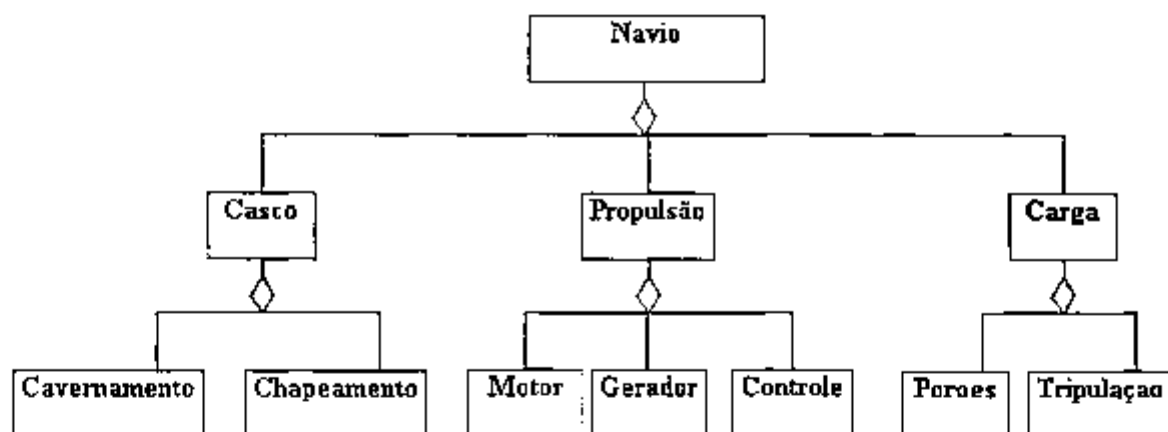


Figura 32 - Diagrama de classes do domínio do produto

#### 4.3.2.2 Análise do domínio do gerenciamento do estaleiro

O domínio do gerenciamento trata do planejamento e controle do desenvolvimento do projeto. As relações existentes entre os gerentes (estaleiro) e o armador (cliente), são representadas pela classe contrato. O contrato reflete, em um nível alto de abstração, os requisitos do produto e as obrigações do estaleiro. O contrato é modelado como o qualificador da

relação entre o armador e a gerência do estaleiro, representada a seguir:

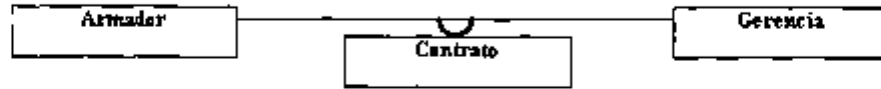


Figura 33 - Diagrama de classes do domínio do gerenciamento

O processo de gerenciamento de um projeto pode ser entendido, simplificado, pelo diagrama de eventos da figura Figura 16 . Nele observa-se que o gerenciamento, assim como a maioria dos processos em um SCAPE tem uma característica cíclica.

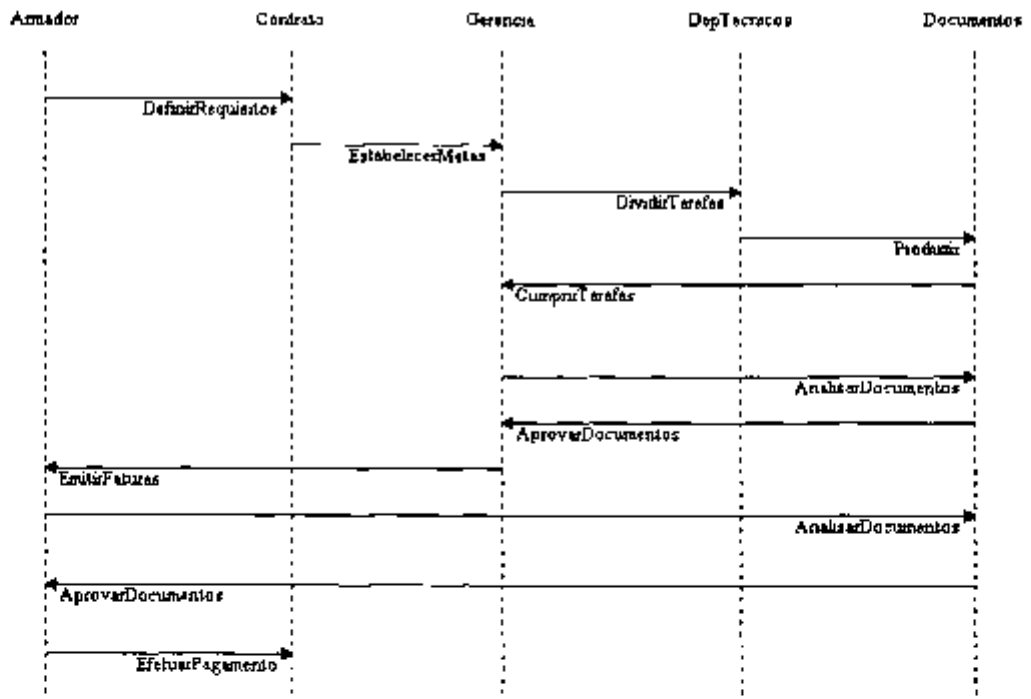


Figura 34 - Diagrama de eventos do processo de gerenciamento

#### 4.3.2.3 Análise do domínio do desenvolvimento do navio

O domínio do desenvolvimento define as relações entre as equipes de projeto e os modelos desenvolvidos sobre o produto. O diagrama de classes que descreve a equipe técnica se

assemelha ao organograma do departamento técnico do estaleiro. Nele estão representados as diversas especialidades necessárias para o desenvolvimento do projeto do navio, agrupadas em divisões. O diagrama apresenta ainda os modelos de projeto desenvolvidos por esta equipe.

Durante o desenvolvimento do projeto automatizado de um navio, ele é representado por uma maquete eletrônica. Analogamente às maquetes convencionais, o modelo gráfico tridimensional permite extrair desenhos de diversos tipos e realizar estudos de interferência e montagem. Pode-se estabelecer uma relação entre a decomposição do produto e os modelos necessários para representá-lo. Um modelo matemático especial, o modelo de estabilidade, auxilia os projetistas na análise da estabilidade hidrostática da embarcação, e também permite estimar seu desempenho hidrodinâmico.

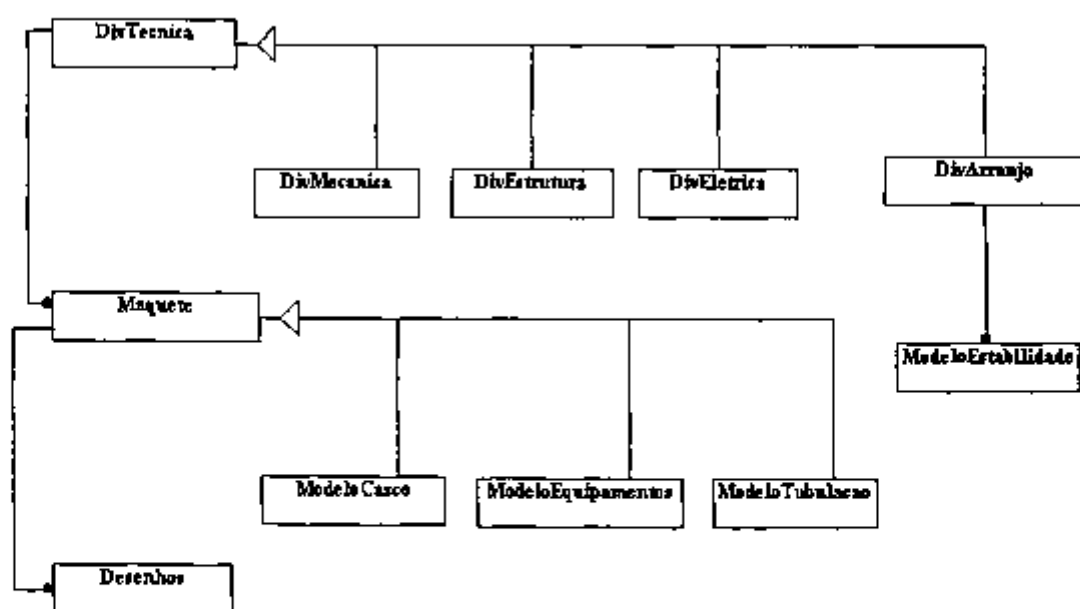


Figura 35 - Diagrama de classes do domínio do desenvolvimento

Passa-se então a analisar os processos que ocorrem neste domínio. Cada processo é representado por eventos entre as classes do modelo. Os eventos estão relacionados às operações destas classes. Os diagramas de evento que se seguem, descrevem apenas alguns processos principais, mas são suficientes para o levantamento das funções mais importantes no projeto.

No diagrama abaixo descreve-se a concepção do casco a partir das dimensões principais estabelecidas em contrato. As formas principais e a divisão de compartimentos são definidas, no início do processo pela divisão de arranjo. Posteriormente, a divisão de estruturas realiza o projeto estrutural do casco e dos demais compartimentos.

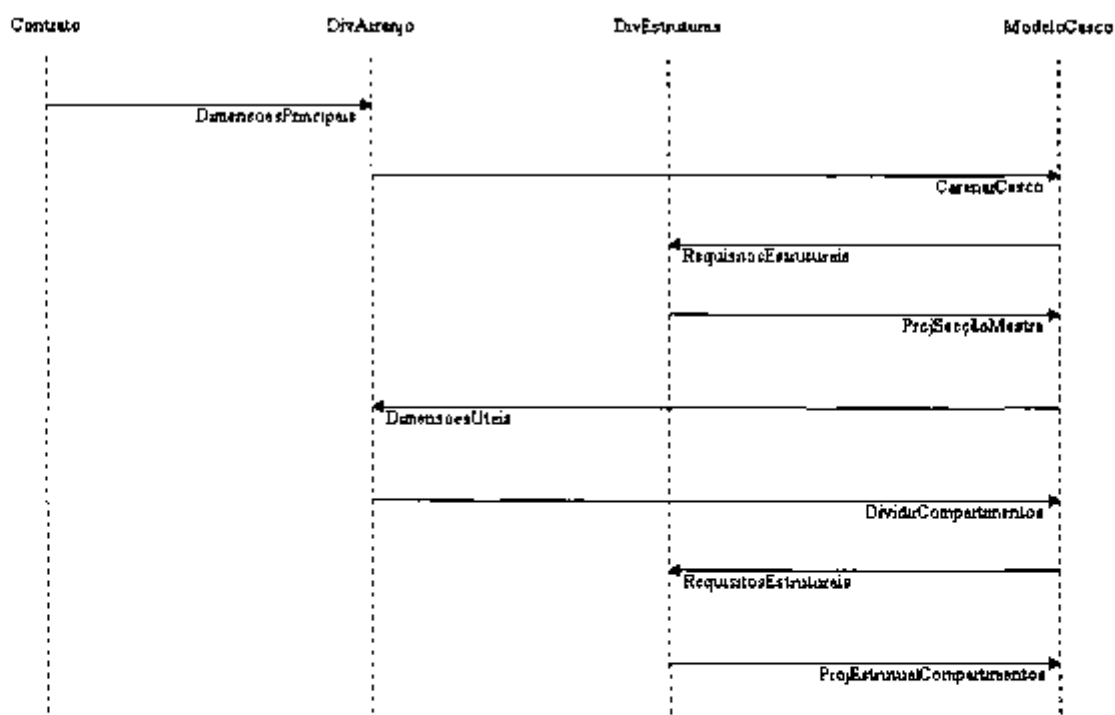


Figura 36 - Diagrama dos eventos de projeto do casco

O diagrama de eventos que se segue sintetiza o projeto de equipamentos mecânicos e elétricos da propulsão do navio.

Parte-se dos requisitos contratuais para se selecionar a propulsão mais adequada ao navio, que é incluída no modelo de equipamentos. As atividades são concluídas estabelecendo-se um arranjo preliminar para a sala de máquinas.

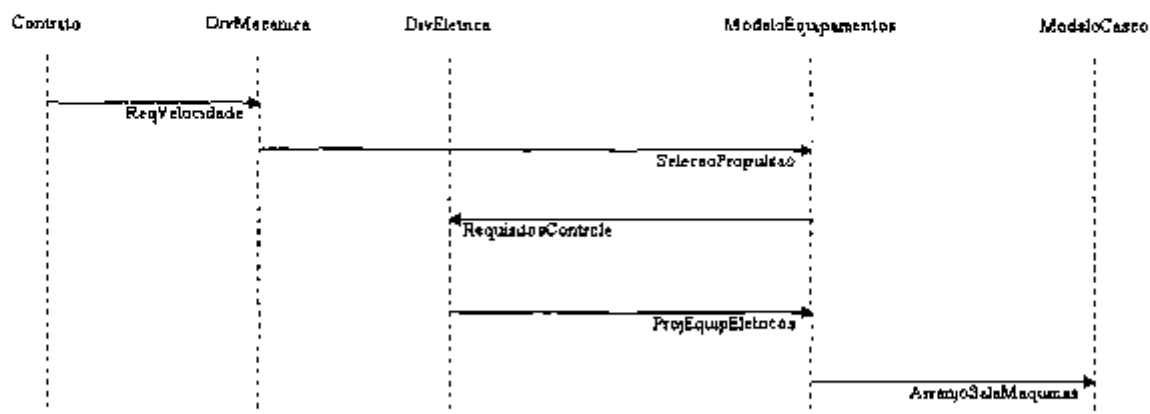


Figura 37 - Diagrama de eventos do projeto mecânico e elétrico

A figura 16 representa o processo de produção dos desenhos isométricos de tubulação para a fabricação e montagem, e dos diagramas unifilares para a instalação elétrica.

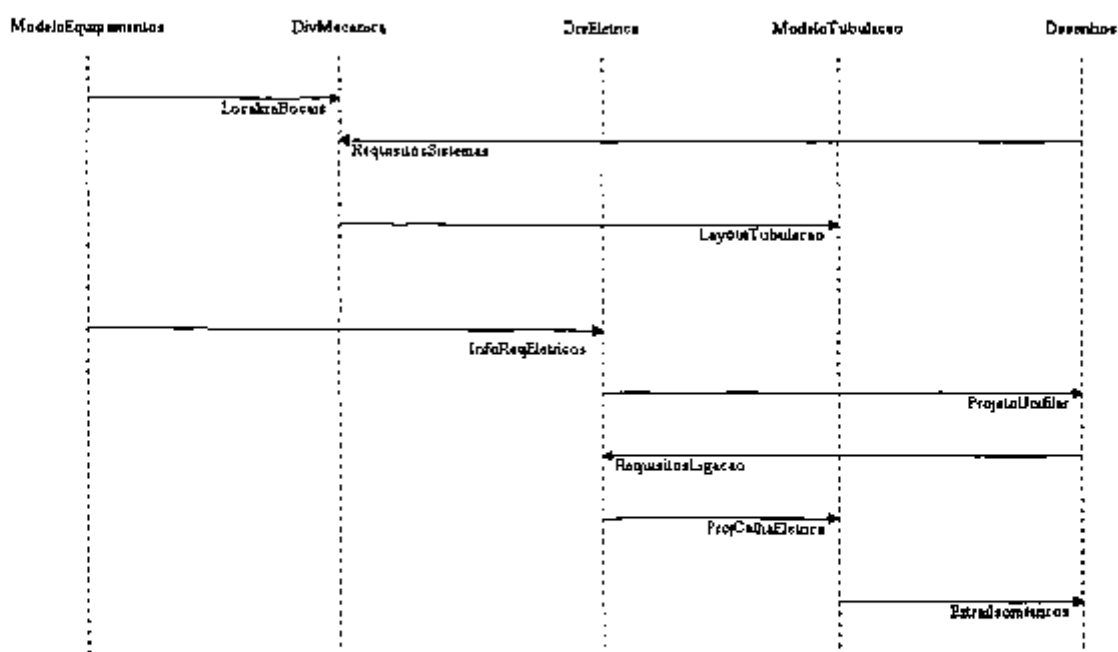


Figura 38 - Diagrama de eventos do projeto de tubulação

Completada a maquete eletrônica pode-se analisá-la quanto a existência de interferência entre os elementos estruturais, equipamentos e tubulação. Os erros devem ser negociados e corrigidos entre as divisões técnicas. As curvas do casco e a lista do peso dos equipamentos são utilizados para a avaliação do desempenho da embarcação sob o ponto de vista hidrostático. Estes processos estão representados no diagrama de eventos abaixo:

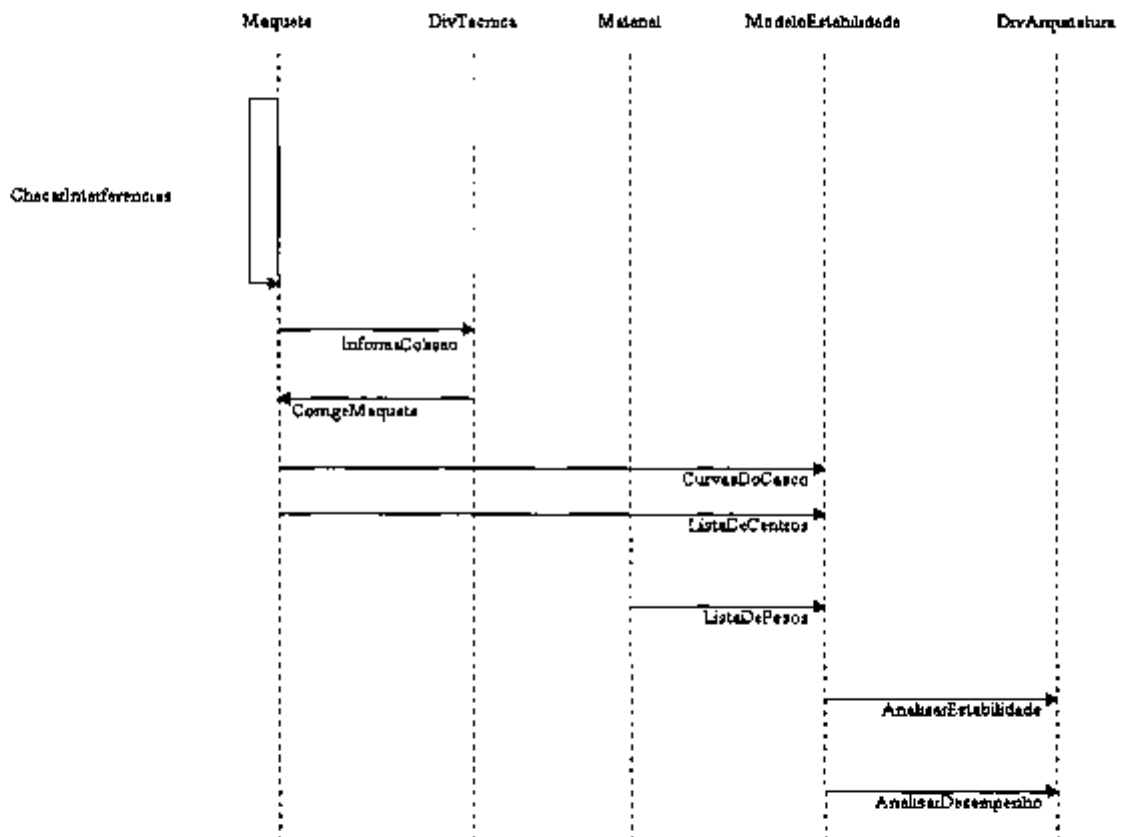


Figura 39 - Diagrama de eventos da análise da maquete

#### 4.3.2.4 Análise do domínio da infra-estrutura

Um projeto deste porte produz uma grande quantidade de informação na forma de documentos de projeto, além de extensas listas de material que devem ser controladas. A divisão da qualidade encarrega-se de padronizar o material a ser

empregado na construção, e de administrar o conjunto de documentos produzidos, como representa os seguintes diagramas:

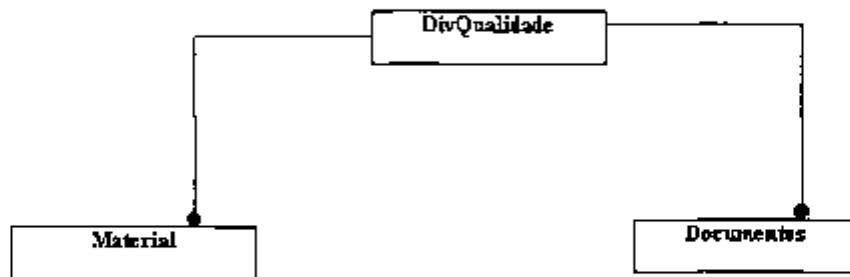


Figura 40 - Diagrama de classes do domínio da infra-estrutura

Entre estas classes ocorrem os processos de gerenciamento dos documentos e das listas de materiais, que são retratados a seguir:

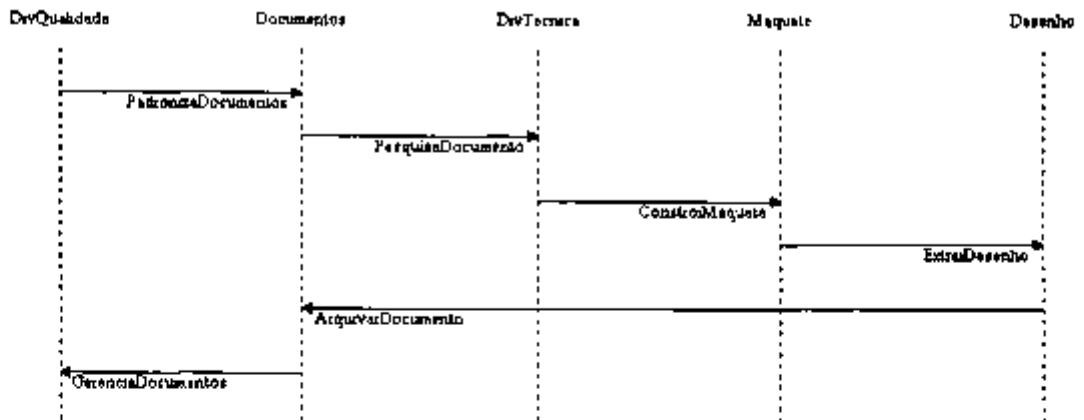


Figura 41 - Processo de gerenciamento dos documentos de projeto

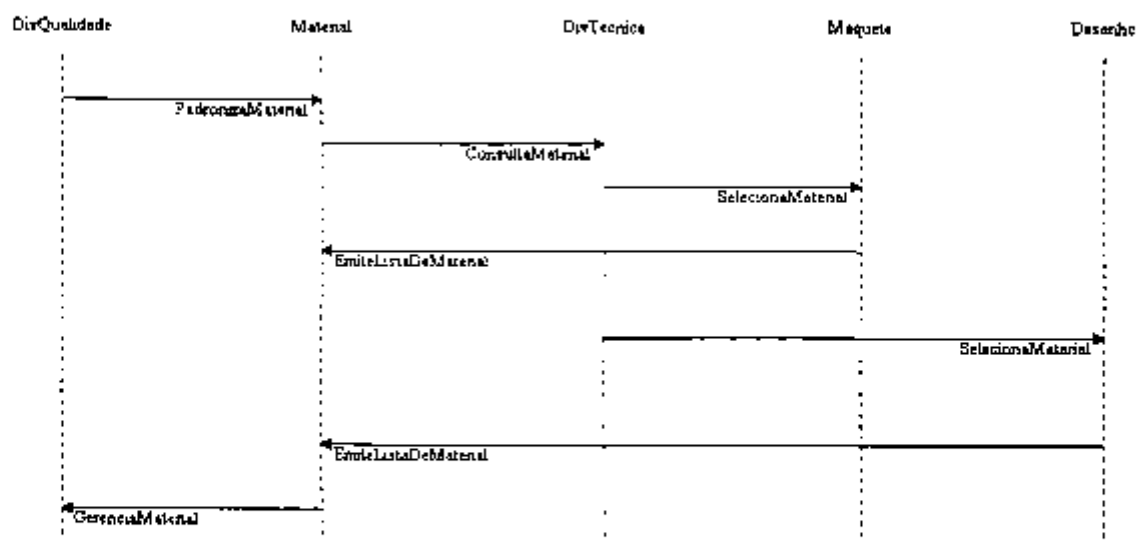


Figura 42 - Processo de gerenciamento das listas de material

Conclui-se a análise quando há, por parte do projetista, um suficiente entendimento dos processos a serem automatizados. Nesta fase integram-se os diversos diagramas de eventos ao diagrama de classes, através da distribuição de operações que disparam os eventos. O diagrama de classes resultante resume o resultado da análise de domínios, e alimenta a fase de projeto do SCAPE.

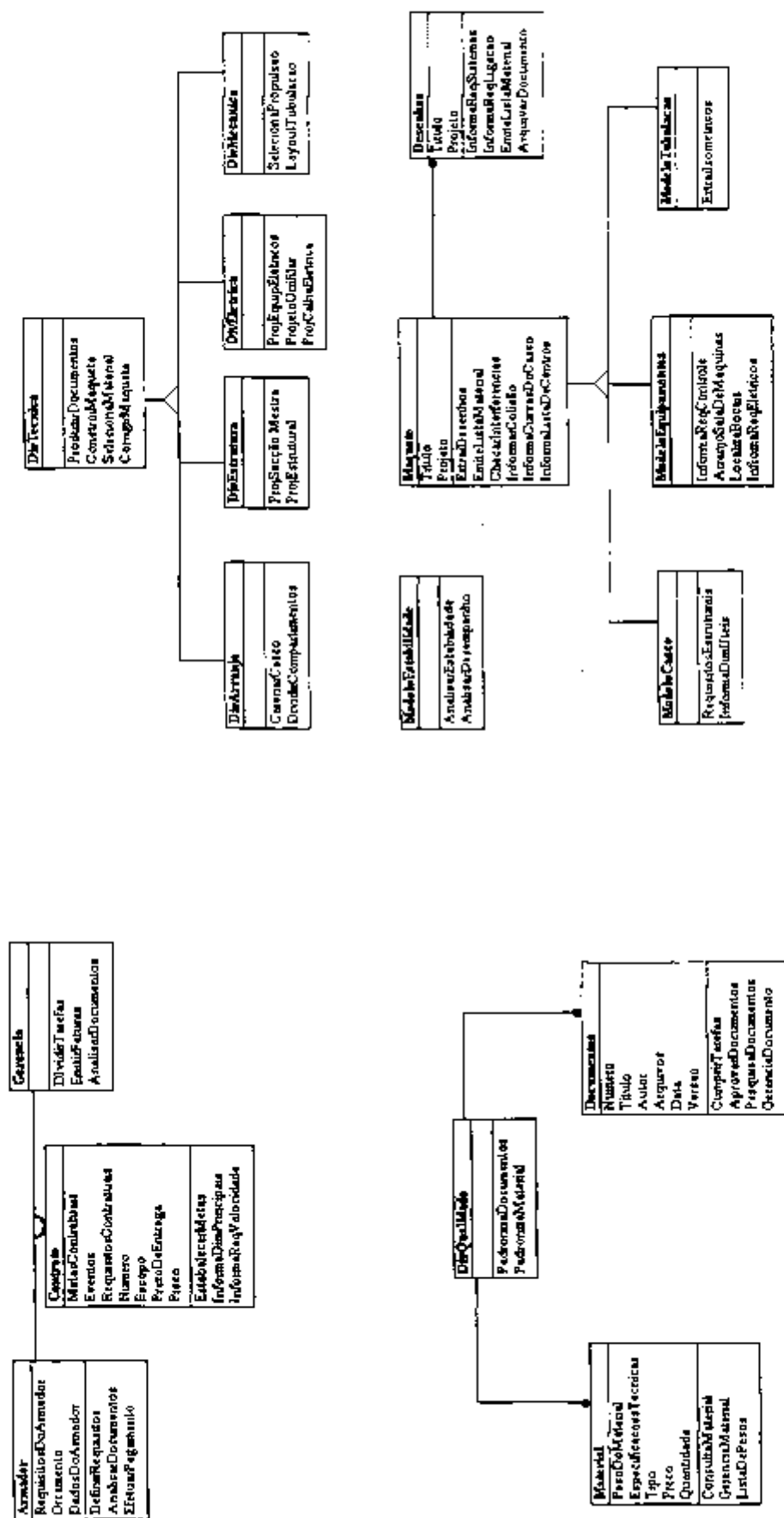


Figura 43 - Diagrama de classes resultante da análise de domínios

### 4.3.3 Projeto do SCAPE do caso 2

Passa-se então a editar o diagrama de classes com objetivo de incorporar a ele as definições que permitam a sua tradução, ao final, em produtos de software. Procura-se criar um diagrama equilibrado, redistribuindo a responsabilidade de cada evento entre as operações das classes envolvidas, procurando não aumentar o acoplamento entre as classes. Procura-se escolher os nomes para identificar as operações de modo a relacioná-las com os eventos.

Distribuídas as operações, são identificadas as classes que devem ser traduzidas em software. Neste exemplo não serão traduzidas as classes que identificam as equipes de projeto, a divisão da qualidade tão pouco a gerência. Esta decisão se baseia no fato que se procura um sistema que de apoio ao projetista e a gerência, e não automatizar todas suas funções. Estas classes fazem parte do modelo do sistema mas representam o papel dos personagens humanos no processo, e não de componentes de software. As funções destas classes que deverão ser automatizadas podem ser distribuídas entre as outras classes do sistema, que são implementadas em software e que interagem com estes personagens. A classe armador, apesar de representar um correspondente humano, poderia ser implementada para representar a interface do cliente com o SCAPE.

Uma vez identificadas as classes que serão implementadas em software deve-se classificar as suas operações conforme a lista de funções padronizadas de um SCAPE. Esta tarefa também

é realizada com apoio do programa CASE. Escolhe-se um classe de funções padronizadas para cada operação, comparando-se a especificação desta operação com a descrição de cada função. Caso uma operação possa ser categorizada como duas ou mais funções, deve-se investigar se esta operação não pode ser decomposta em operações mais simples, pois este fato pode indicar que o modelo precisa ser mais detalhado.

Nesta fase definem-se a arquitetura de processamento do SCAPE. Propõe-se construir o SCAPE sobre uma arquitetura cliente/servidor, onde os postos de trabalho dos clientes se integram à servidores dedicados a funções específicas através de uma rede de comunicação de dados. Os postos de trabalho podem ser baseados em Workstations RISC ou Microcomputadores INTEL, dependendo da disponibilidade de produtos nestas plataformas. Não serão considerados os problemas de comunicação de dados envolvidos neste exemplo, nem maiores considerações sobre o hardware, supondo-se que todo o sistema é processado em uma rede local e exclusiva para este fim, que será motivo de uma configuração posterior. A definição do tipo, configuração e quantidade do hardware deverá ocorrer após a definição do software.

Concluída a redistribuição das operações, a classificação das funções e as definições da arquitetura, temos o diagrama de classes final do sistema, mostrado a seguir.

#### 4.3.4 Implementação do SCAPE no caso 2

O resultado do projeto é utilizado para alimentar a etapa final de implementação realizada pelo programa BSCAPE, conforme o procedimento de uso descrito no anexo D. Procura-se no banco de produtos de software atender aos requisitos de cada classe, analisando o conjunto de funções que deve ser desempenhada pela classe, em cada área de aplicação.

A inclusão das áreas de aplicação na consulta serve para limitar a busca, levando a um conjunto de programas que ao mesmo tempo executa a função de projeto, mas atende a uma área de aplicação específica. Cada consulta ao banco resulta em um conjunto de produtos candidatos a implementar a classe, e que devem ser analisados um a um frente às necessidades do sistema.

A configuração descrita pela tabela, foi resultado da procura dos produtos no banco para atender as classe do diagrama de projeto. Procurou-se manter como critério de seleção um número pequeno de fornecedores, o que levou a escolher a quase totalidade de produtos do mesmo fabricante.

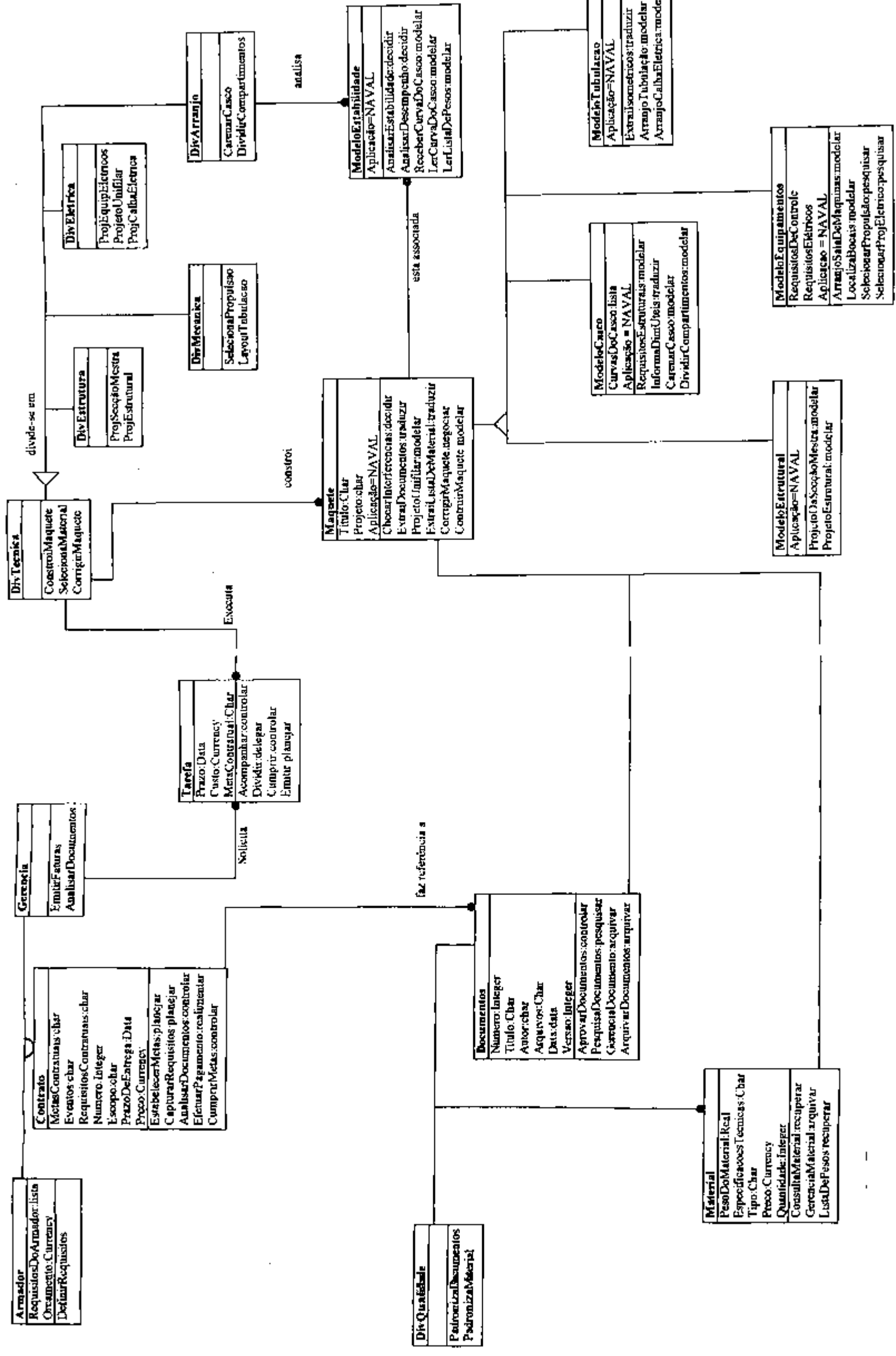


Figura 44 - Diagrama de classes resultante do projeto

Classes	Produto Fabricante	Descrição
Contrato Tarefa	Project  Microsoft	Gerenciador de projetos para facilitar a montagem e supervisão de atividades. Auxilia a colocar em prática os planos, comunicá-los a outras pessoas e administrar as modificações
Material	Material takeoffs  Intergraph	Modulo do I/VDS que extrai listas de material diretamente do modelo, em qualquer fase do projeto.
Documentos	Microstation TeamMate  Bentley	Ferramenta para gerenciamento de documentos e workflow, integrada ao Microstation. Compatível com formato DGN e também de outros formatos. Os documentos são agrupados em "sets" e organizados por projeto.
Maquete	I/VDS  Intergraph	Intergraph's Vehicle Design System (I/VDS) constitui o módulo central de um conjunto de produtos que apoia o projeto, construção e manutenção de embarcações e outros tipos de veículos. O programa desenvolve um modelo tridimensional do veículo que pode ser simulado e analisada as interferências entre seus componentes.
Maquete	Microstation Bentley	Programa de modelagem gráfica, compatível com o I/VDS que permite extrair e elaborar desenhos bidimensionais para compor os documentos de projeto.
Modelo Estabilidade	FastShip  Proteus	Programa para projeto de casco, acessórios e superestrutura de navios. Usando NURBS da modelagem preliminar até o ajuste final parametrizado. O projetista esculpe uma superfície interativamente, e observa o casco em tempo real. O programa calcula as curvas hidrostáticas. O casco pode ser parametrizado sendo o LCB, coeficiente prismático, bloco etc.
Modelo Casco  Modelo Estabilidade	GHS  Creative Systems inc	programa de atua em todos os aspectos de geração de curvas do casco, flutuação, resistência longitudinal aplicada a navios de superfície e submarinos.
Modelo Casco  Modelo Estrutural	I/LOFT  Intergraph	Intergraph's Vehicle Lofting System (I/LOFT) integrado ao I/VDS permite calcular reforçadores e placas estruturais para fabricação.

Modelo Estrutural	I/STRUCT Intergraph	I/STRUCT módulo de projeto de elementos estruturais com interface para programas de análise e fabricação.
Modelo Equipamentos	Piping	módulo do I/VDS que permite a seleção e projeto de tubulações pela seleção em catálogos on-line. O módulo permite analisar a consistência da linha de tubulação quanto a especificações e interferências.
Modelo Tubulação	Intergraph	
Modelo Tubulação	I/ROUTE Intergraph	Módulo do I/VDS que permite definir os caminhos de tubulação, Ar-condicionado e caminhos para cabeação elétrica. A tubulação é definida em conjunto com suas especificações de material e de modo a garantir continuidade.

Tabela 9 - Lista da configuração proposta

Selecionada a configuração de software deve-se distribuir este software entre as estações de trabalho e servidores seguindo os requisitos de cada produto. Estes equipamentos devem ser interligados por uma rede de comunicação de dados. Objetiva-se nesta fase garantir desempenho e disponibilidade aos projetistas. O projeto de hardware e da rede não fazem parte do escopo deste trabalho.

O diagrama que se segue apresenta o SCAPE completo incluindo o domínio de implementação, representado pelos produtos de software selecionados para cada classe.

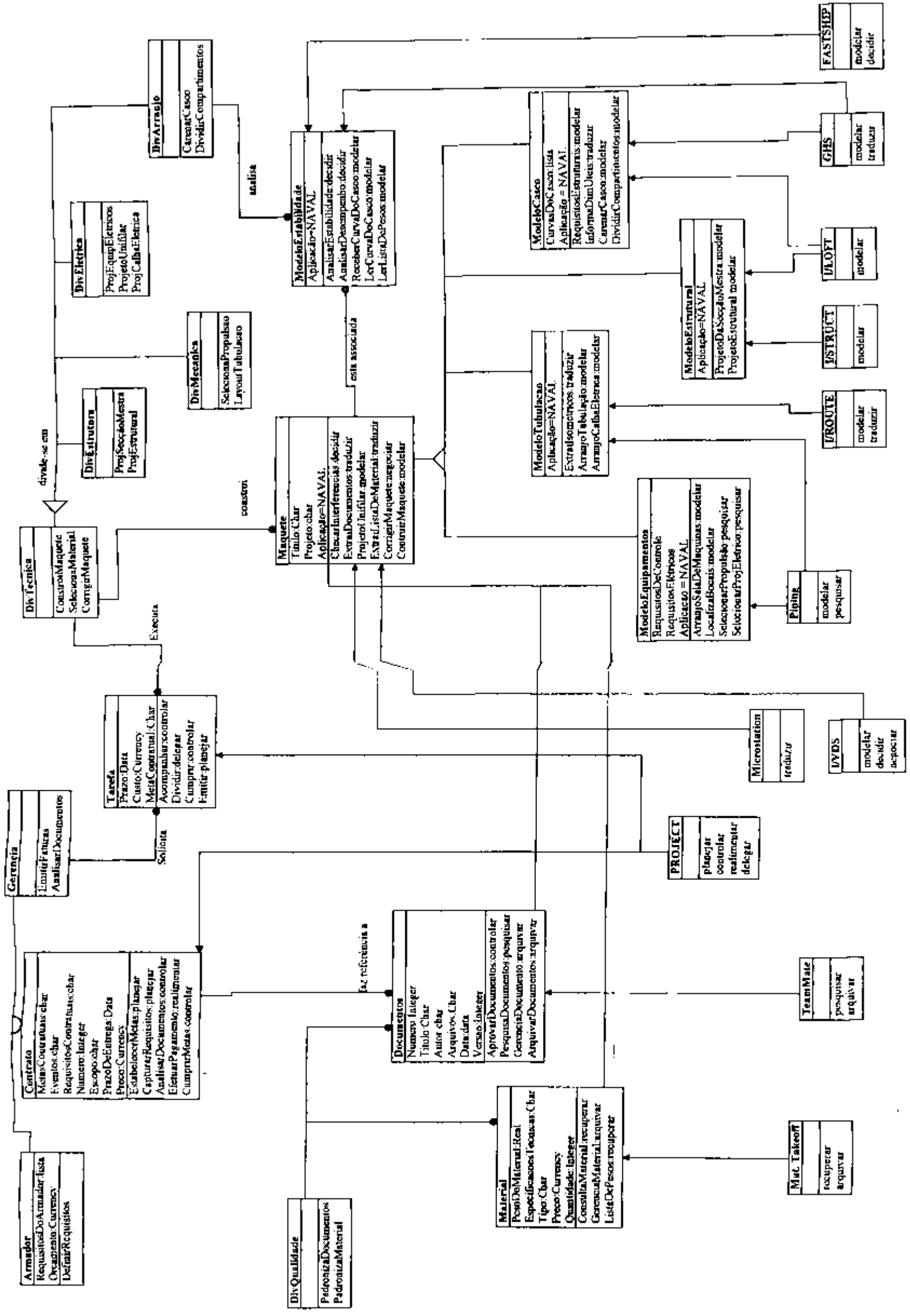


Figura 45 - Diagrama de classes final do SCAPE no caso 2.

JOSÉ EDUARDO ZINDEL DEBONI

UM MÉTODO PARA INTEGRAÇÃO DE SISTEMAS DE  
APOIO A ENGENHARIA ATRAVÉS DA REUTILIZAÇÃO  
DE SOFTWARE

Tese apresentada à Escola  
Politécnica da Universidade de  
São Paulo para obtenção do  
título de Doutor em Engenharia

Área de Concentração:  
Engenharia de Computação e  
Sistemas Digitais

Orientador:  
José Sidnei Colombo Martini

1997



#### **4.4 Caso 3: Sistema computacional de apoio ao projeto de propulsão nuclear**

Este exemplo de aplicação baseia-se no caso real da criação do SCAPE na área de propulsão nuclear do Centro Tecnológico da Marinha em São Paulo (CTMSP). Inicialmente avalia-se o estágio atual da automação de projetos usando o questionário de apoio a análise, para, a seguir, propor-se alterações ao SCAPE de modo a aproximá-lo de uma situação ideal, através do método de integração.

##### **4.4.1 Particionamento dos domínios do caso 3**

É possível fazer um paralelo entre a estrutura dos domínios do SCAPE e a organização funcional do CTMSP, associando-se cada superintendência da estrutura organizacional com um domínio. Esta comparação facilita o entendimento do sistema como um todo e isola a responsabilidade de cada grupo, como mostra a figura 46:

- a) considera a existência de um domínio da industrialização, responsável pela construção dos produtos projetados;
- b) chama o domínio da infra-estrutura de domínio de apoio administrativo;
- c) chama o domínio do desenvolvimento de domínio técnico.

O produto (reator nuclear) impõe uma estrutura gerencial própria para tratar das peculiaridades das suas componentes. Por se tratar de um projeto complexo, nem todas as suas partes são projetadas por este SCAPE, alguns componentes são produtos vindos de outros sistemas de projeto externos, que ao final devem ser incorporados ao produto.

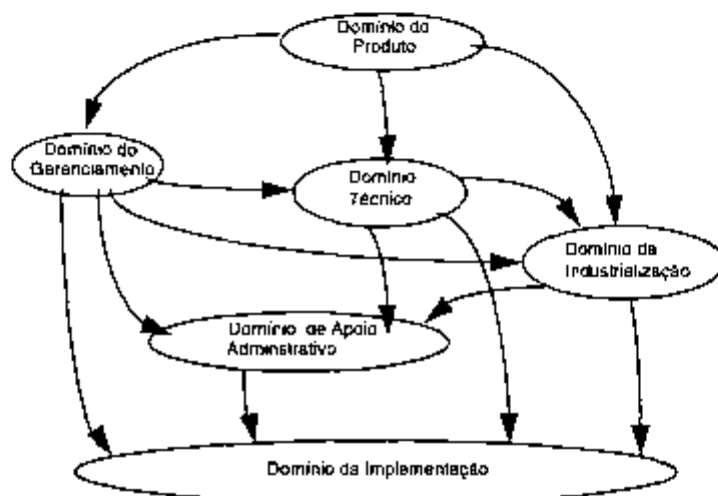


Figura 46 - Diagrama de domínios

Cabe ao domínio técnico fornecer ao domínio do gerenciamento as informações técnicas para o desenvolvimento interno do projeto, ou para validar o projeto desenvolvido externamente. O domínio de apoio administrativo mantém a infra-estrutura de comunicação e o gerenciamento da obtenção do material para a produção. A industrialização cuida da fabricação e montagem dos componentes projetados.

Para o estudo do SCAPE são analisados apenas os componentes dos domínios que participam da fase de projeto. Assim, não são considerados neste trabalho o domínio da industrialização e a parcela dos demais domínios que atendam exclusivamente a produção.

#### 4.4.2 Análise do questionário de apoio à análise no caso 3

Para se avaliar, ainda que preliminarmente, o estágio de automação do SCAPE, utilizou-se do questionário de apoio à análise resultando no quadro de respostas apresentado a seguir.

Estágio de Automação:		inicial	intermédio	avanzado	totalmente
Gerenciamento	Planejamento	1	2	3	4
	Delegação	5	6	7	8
	Controle do	9	10	11	12
Desenvolvimento	Realimentação	13	14	15	16
	Pesquisar por dados	17	18	19	20
	Concepção e	21	22	23	24
	Tomada de decisão	25	26	27	28
Infra-estrutura	Negociação	29	30	31	32
	Arquivamento	33	34	35	36
	Comunicação	37	38	39	40
	Recuperação	41	42	43	44
	Captura de Dados	45	46	47	48
	Impressão	49	50	51	52
	Tradução	53	54	55	56
	Padronização	57	58	59	60
Treinamento	61	62	63	64	

*Foram marcadas as respostas positivas*

Figura 47 - Quadro de respostas do questionário de análise

As respostas foram obtidas através de uma inspeção ao CTMSP. O resultado reflete um bom nível geral de automação de projeto, em especial no domínio do desenvolvimento. Esta ênfase da automação no desenvolvimento é explicada pela complexidade intrínseca à área nuclear, que demanda um uso intenso do computador.

Como resultado da análise prévia, observa-se o que deve ser investigado com mais detalhe e onde concentrar as alterações eventuais no SCAPE existente para aumentar seu nível de automação. Comparando-se o número de respostas positivas ao questionário, caracteriza-se que os domínios de apoio administrativo e de gerenciamento possuem, no presente momento, um menor grau de automação quando comparado com o domínio do desenvolvimento.

Analisando-se as funções que geram respostas negativas, observa-se os processos de realimentação do gerenciamento, e de captura e recuperação das informações de projeto devem ser investigados com mais detalhe pois podem levar a uma maior automação.

#### **4.4.3 Análise dos domínios do caso 3**

A análise dos domínios é feita através dos diagramas de eventos para os principais processos que ocorrem no SCAPE; que ajudam a definir as classes e as comunicações trocadas entre elas.

#### 4.4.3.1 Análise do domínio do Produto

Um reator nuclear de água pressurizada é formado pelos seguintes componentes principais: vaso de pressão, núcleo combustível, ciclo primário, gerador de vapor, ciclo secundário, grupo turbo-gerador. Cada componente pode ser decomposto em um nível maior de especialização segundo as áreas de engenharia envolvidas. Os componentes do produto se relacionam entre si, conforme o diagrama abaixo.:

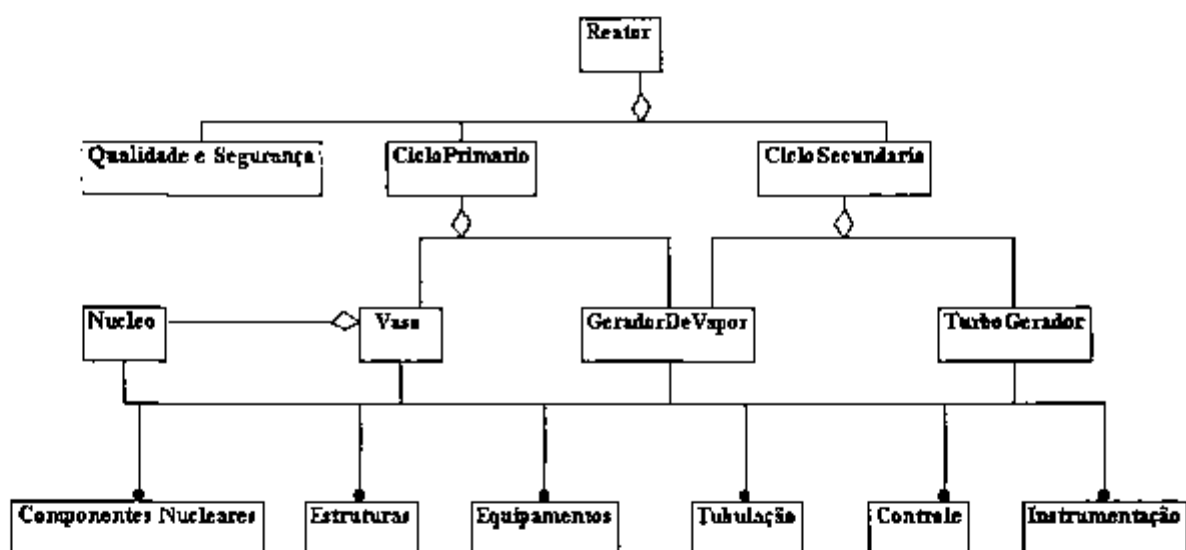


Figura 48 - Diagrama de classes do domínio do produto

#### 4.4.3.2 Análise do domínio do gerenciamento

O domínio do gerenciamento trata do planejamento e controle do desenvolvimento do projeto. Um projeto deste porte não pode ser totalmente realizado internamente. Assim, o gerenciamento se divide em atividades de gerenciamento interno para elaboração e análise da documentação técnica do projeto, e atividades ligadas ao gerenciamento de

fornecedores externos. No gerenciamento externo o objeto de controle é o pedido de suprimento (PESUP), e no interno é a atividade delegada às divisões técnicas, que recebe, internamente, o nome de TAREFA. Estes dois itens de controle são componentes de um elemento macro de planejamento chamado PACOTE. O cliente, no caso a Marinha do Brasil, estabelece metas associadas ao orçamento do empreendimento em um planejamento básico de alto nível. A figura abaixo modela estas relações:

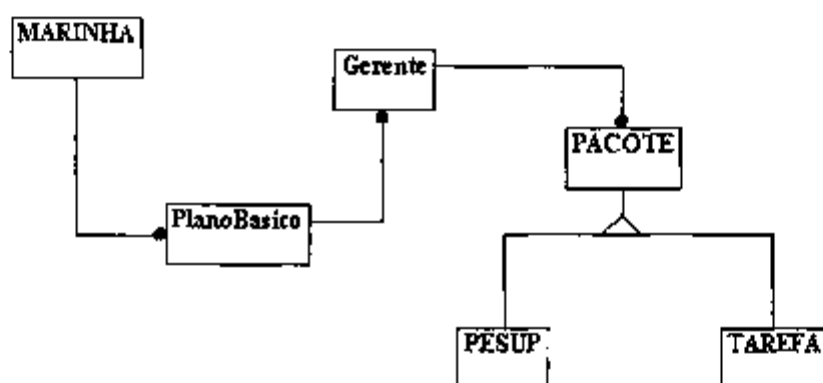


Figura 49 - Diagrama de classes do domínio do gerenciamento

O processo de gerenciamento de um projeto pode ser entendido pelo diagrama da figura 50, onde se observa o ciclo de planejamento e controle com maior detalhe.

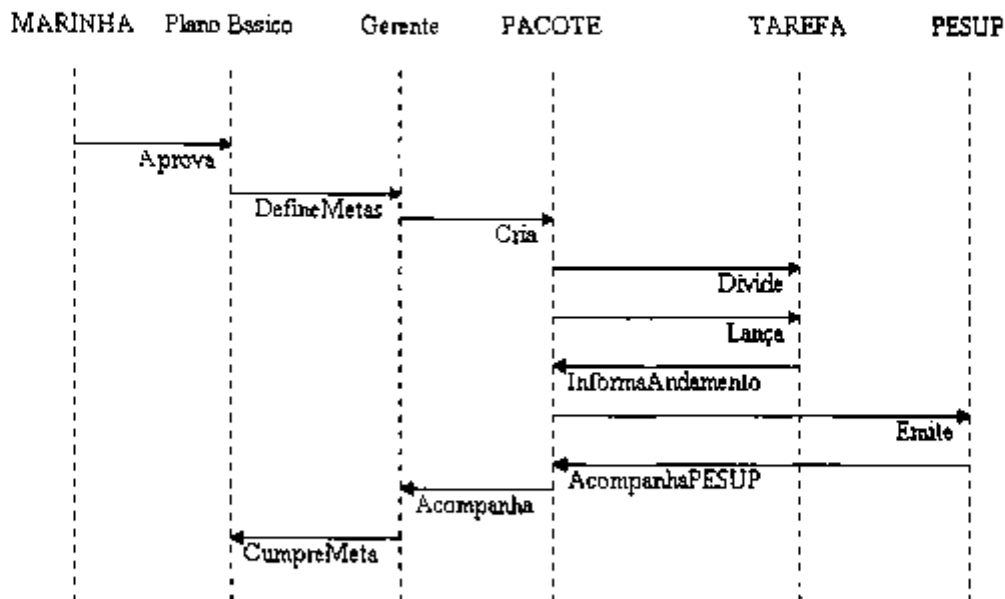


Figura 50 - Diagrama de eventos do processo de gerenciamento

#### 4.4.3.3 Análise do domínio técnico

No domínio técnico encontram-se as relações existentes entre as equipes de projeto e os modelos matemáticos e gráficos desenvolvidos sobre o produto. Os modelos desenvolvidos pelas equipes técnicas podem ser classificados em dois grandes tipos: maquete eletrônica e modelos matemáticos de análise. Diversos modelos destes tipos são desenvolvidos para cada especialidade envolvida no projeto. Os processos que ocorrem no domínio técnico seguem, normalmente, um ciclo de concepção e análise. A maquete eletrônica é utilizada para concepção do produto enquanto a análise é realizada sobre modelos matemáticos. Os resultados da análise são registrados em memoriais de cálculo, enquanto que se extrai da maquete eletrônica desenhos e listas de material.

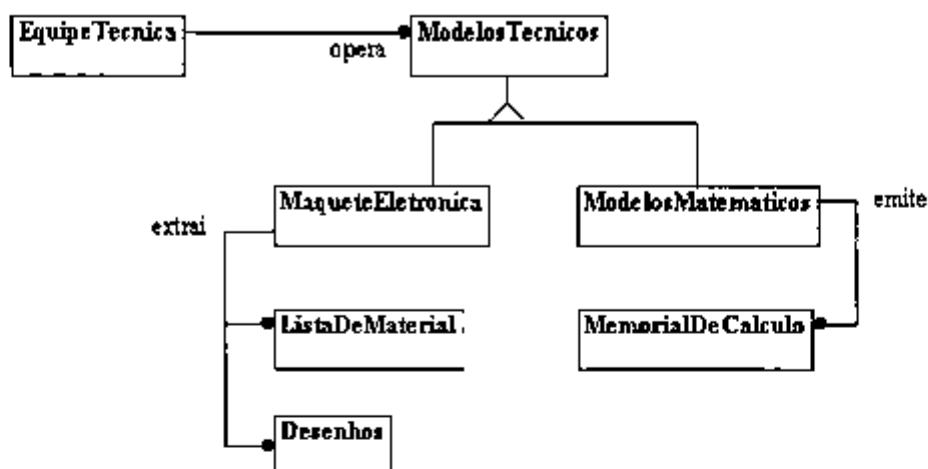


Figura 51 - Diagrama de classes do domínio do desenvolvimento

#### 4.4.3.4 Análise do domínio de apoio administrativo

No domínio de apoio administrativo estabelece-se a infraestrutura para os domínios de gerenciamento e técnico. Os elementos principais neste domínio são o documento técnico que se divide em relatórios técnicos e documentos de engenharia, e o material.

Os relatórios técnicos são compostos por memoriais de cálculo, documentando os estudos de análise e comportamento realizados. Os documentos de engenharia são documentos formais que descrevem uma parte do produto para a produção, e se compõem de desenhos e listas de material. Ambos os documentos são extraídos dos modelos desenvolvidos pelo domínio técnico. Os documentos de engenharia são extraídos da maquete eletrônica, enquanto os relatórios técnicos, na sua maioria, dos modelos matemáticos.

Outra classe importante da infra-estrutura é a classe material, que representa todos os componentes do produto, representado na sua unidade para a construção. Os materiais são adquiridos pelo domínio administrativo segundo documentos de engenharia do tipo: especificação técnica.

A classe aquisição representa a unidade organizacional responsável pela obtenção do material. A classe Arquivo Técnico, formalmente pertencente ao domínio técnico, responde pelo gerenciamento dos documentos técnicos.

O diagrama de classes neste domínio está esquematizado pela figura 52:

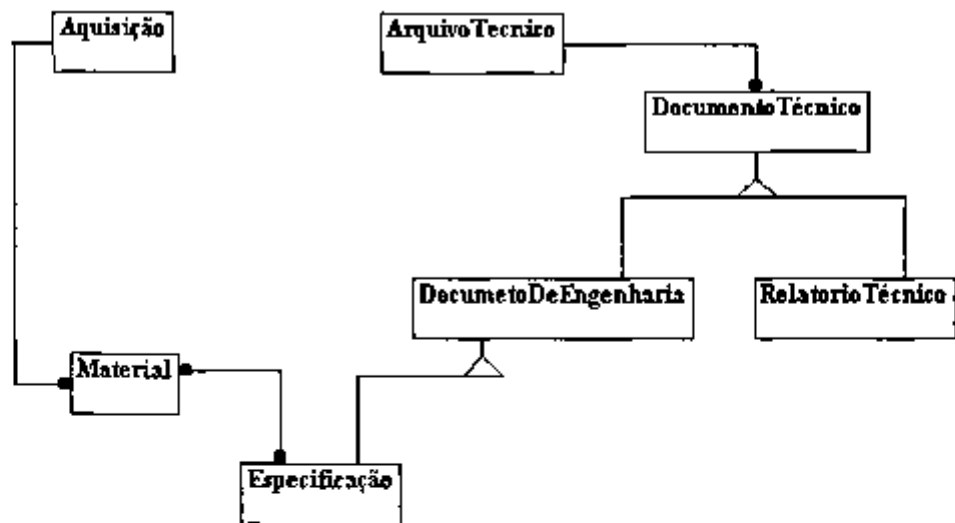
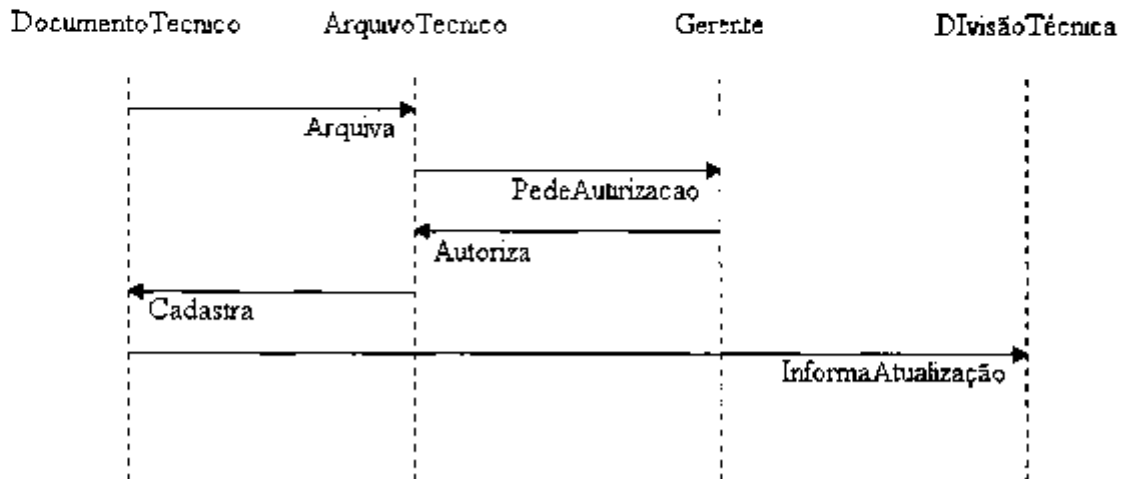


Figura 52 - Diagrama de classes do apoio administrativo

Com base do conjunto de documentos técnicos e no material adquirido, o produto pode ser construído, no domínio da industrialização. O diagrama de eventos a seguir mostra algumas das atividades de gerenciamento da informação

técnica. serva-se que participam do processo de gerenciamento dos documentos de projeto classes de todos os domínios.



*Figura 53 - Processo de gerenciamento dos documentos de projeto*

Outros processos são encontrados neste domínio, como os processos de emissão de um documento, revisão de um documento, entre outros, mas que não serão detalhados neste trabalho.

#### **4.4.3.5 Análise do domínio da implementação**

Como se trata de um SCAPE existente, diversos produtos de software podem ser encontrados implementando as classes dos modelos apresentados. Realizado um levantamento destes produtos, eles podem ser incluídos no modelo de objetos ainda na fase de análise. Os produtos de software

encontrados são registrados como classes do domínio da implementação.

A tabela abaixo mostra alguns dos produtos disponíveis atualmente:

Produto	Descrição
SISUP	Sistema de informações gerenciais sobre os pedidos de suprimentos (PESUP) e de apoio os processamento de aquisições de material. Composto pelos módulos RQ - requisitante, DPC - planejamento e controle, AQ - aquisição e MT - material.
SIPLADI	Sistema de informações gerenciais sobre os planos básicos da MARINHA
PROJECT	Programa de planejamento
Microstation	Programa CAD para desenho em microcomputadores
PDMS	Programa CAD de gerenciamento de uma maquete eletrônica de uma instalação industrial. Composto pelos módulos: PEGS - desenho isométrico, DESIGN - modelagem tridimensional, PIPING - roteamento de tubulação e STRUC - projeto estrutural.
Design Center	Programa integrado de projeto de placas de circuito impresso
ANSYS	Programa de análise estrutural pelo método dos elementos finitos
Códigos Nucleares	Biblioteca de programas para simulação e análise em engenharia nuclear. Composto por diversos programas específicos.
Pipe Stress	Programa de análise da estrutura de suporte de tubulação pelo método dos elementos finitos
MS_OFFICE	Conjunto de programas composto de editor de texto, planilha de cálculo e programa gráfico de apresentação, uso na preparação dos relatórios técnicos.

Tabela 10 - Produtos de software encontrados no CTMSP

Os diagramas de classe de cada domínio podem ser reunidos em um único diagrama, sintetizando os resultados da análise, e oferecendo uma visão do conjunto da organização

em um elevado grau de abstração. Este diagrama encontra-se na figura 54.

#### **4.4.4 Projeto de modificações no SCAPE do caso 3**

Retomando-se os diagramas de classe de cada domínio pode-se incluir as operações identificadas nos diagramas de eventos, e os produtos de software já existentes para implementar as classes de cada domínio. As figuras 55 a 57 mostram o resultado destas operações para os domínios do gerenciamento, do apoio administrativo e técnico, respectivamente.

No domínio técnico, a maquete eletrônica é o repositório central das informações de projeto e nela estão representados todos os componentes da instalação. Este componentes se dividem em esquemas elétricos e de processo, equipamentos, estrutura de sustentação, tubulação equipamentos eletrônicos (hardware).

Todas estas classes são implementadas por produtos de software, como mostra o diagrama. Inclusive os desenhos extraídos da maquete possuem um produto para automatizá-los.

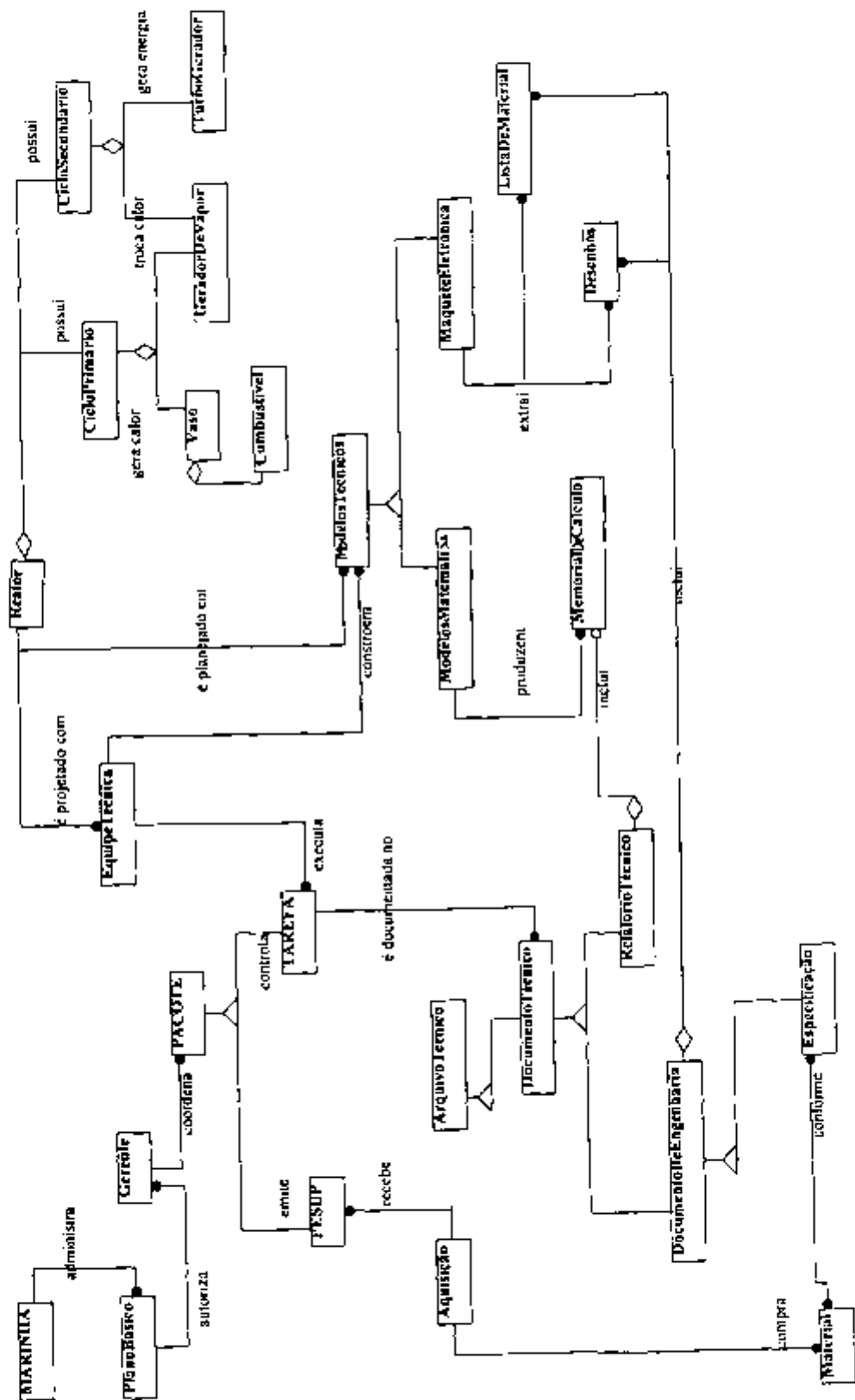


Figura 54 - Diagrama de classes resultante da análise de domínios

Os modelos matemáticos servem para avaliar o comportamento e dimensionar os elementos de projeto. Os modelos matemáticos se dividem em modelos da estrutura e suportes de sustentação, que avaliam o comportamento estrutural e modelos nucleares que avaliam o comportamento nuclear. Este domínio se encontra atendido por um conjunto de produtos de software, que não será modificado nesta fase de projeto.

O domínio do gerenciamento é atendido parcialmente pelos produtos : SIPLADI, SISUP e PROJECT. Porém deve-se observar que as operações da classe TAREFA não são implementadas pelo produto PROJECT, apesar deste produto possuir esta possibilidade. Eventualmente ela pode ser atendidas por um produto complementar, para aumentar o nível de automação deste domínio.

O domínio do apoio administrativo possui a automação das classe de aquisição de material, porém não possui um correspondente produto de software para a classe DocumentoTécnico. Como procura-se aumentar a automação neste domínio, deve-se investigar esta classe e localizar os produtos que possam implementá-la.

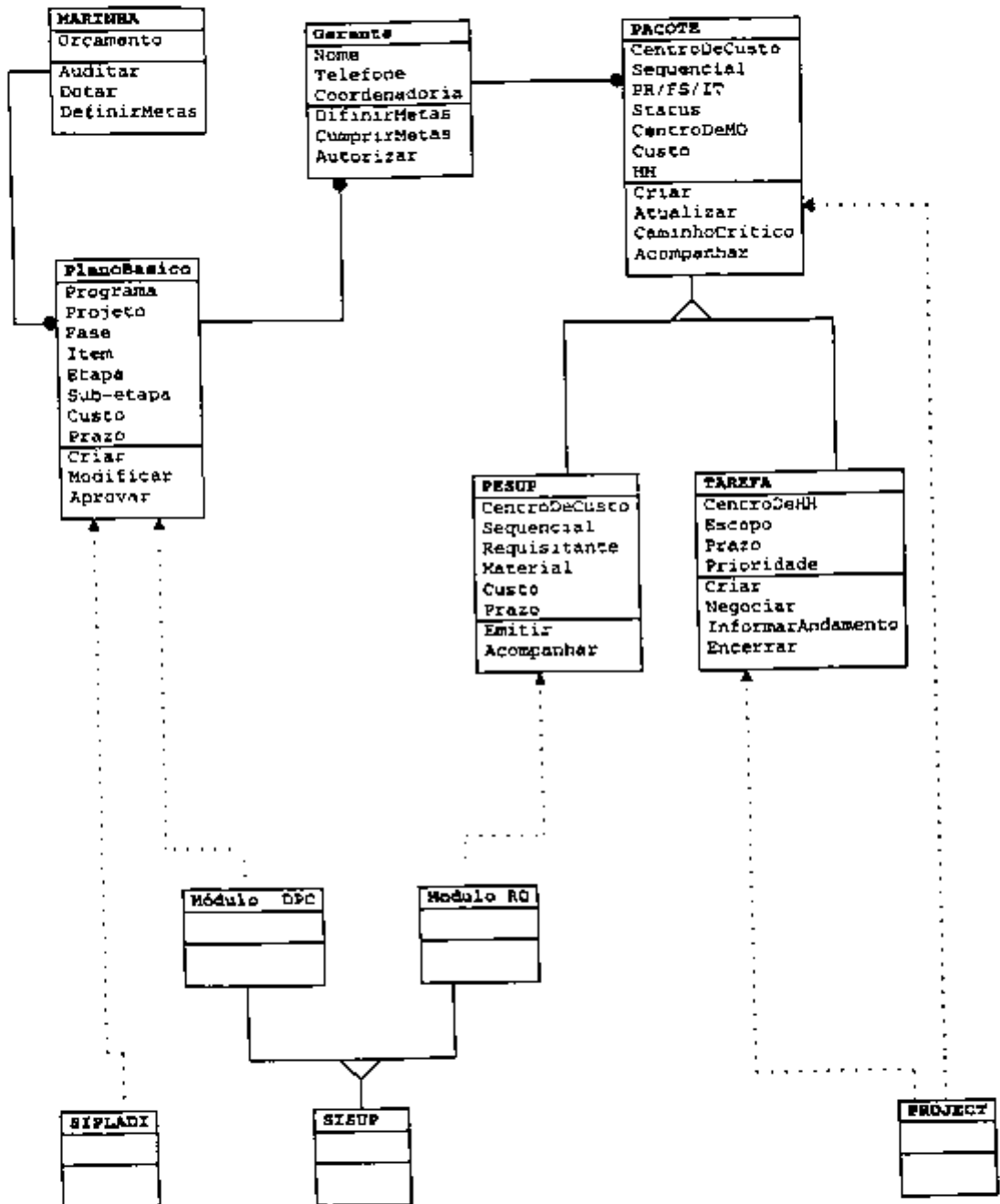


Figura 55 - Diagrama de classes do domínio de gerenciamento na fase de projeto

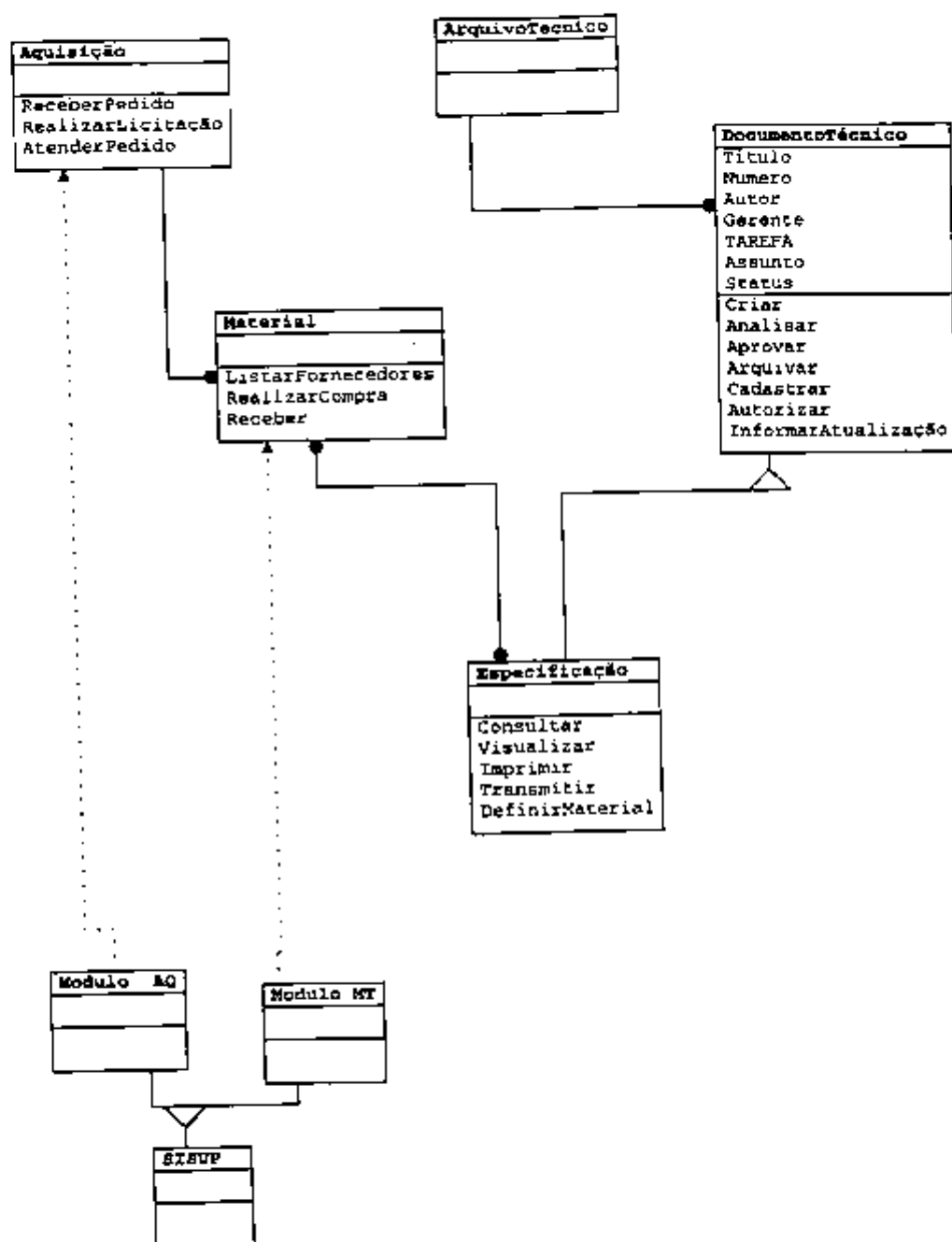


Figura 56 - Diagrama de classe do domínio de apoio administrativo na fase de projeto

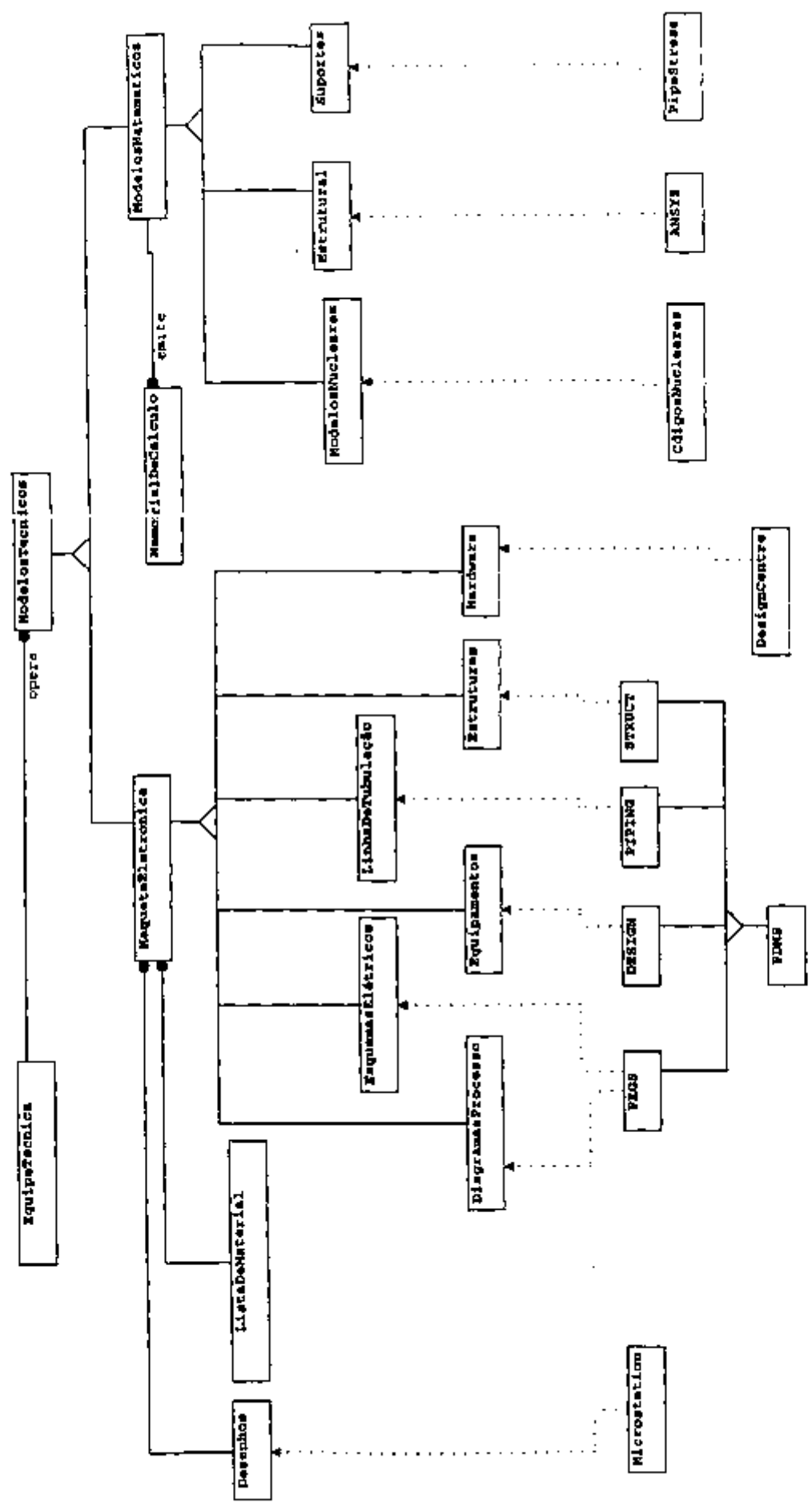


Figura 57 - Diagrama de classe do domínio técnico na fase de projeto

#### 4.4.5 Modificações no SCAPE do CTMSF

Pode-se classificar as operações realizadas por esta classe segundo a arquitetura geral, resultando no diagrama abaixo que é transferido para o programa BSCAPE onde se realiza a implementação.

DocumentoTécnico
Título: texto
Numero: Integer
Autor: texto
Gerente: texto
TAREFA: Integer
Assunto: texto
Status: Integer
Criar: arquivar
Analisar: recuperar
Aprovar: decidir
Arquivar: arquivar
Cadastrar: arquivar
Autorizar: decidir
InformarAtualização: comunicar

*Figura 58 - Classes a serem implementadas no caso 3*

A procura pelos produtos no banco de produtos para atender as necessidades destas classes resulta em um conjunto de opções de produtos, mostrados na tabela a seguir: Com base nesta tabela pode-se selecionar os produtos mais indicados segundo outros critérios técnicos ou e critérios comerciais de fornecimento, que deve indicar o produto final a ser incorporado no SCAPE aumentando a automação do domínio do apoio administrativo.

Produto	Funções	Descrição
Falcom/DMS 5.1	controlar pesquisar arquivar	Gerenciador de documentos para usuários Microstation. Organiza os documentos em diretórios (6 níveis), possui um controle de tempo gasto em cada desenho, para efeito de contabilidade
AM-Workflow	pesquisar arquivar recuperar	Gerenciador de documentos. Desenhos CAD ou qualquer outro documento. Permite a visualização de diversos formatos. Mostra a lista de documentos no formato de cartões ou como uma árvore.
Myriad	recuperar imprimir controlar	Visualizador de arquivos em diversos formatos. Anotação (redlining) de desenhos e documentos. Revisão de documentos. Impressão e Plotagem de documentos eletrônicos.
TeamMate	pesquisar arquivar recuperar	Ferramenta de gerenciamento de documentos e workflow, integrada ao Microstation. Gerencia os formatos DGN entre outros formatos CAD. Os documentos são agrupados em "sets" e organizados por projeto. Possui suporte a bancos de dados relacionais, em redes TCP/IP.
Sightseer 3.21a	Comunicar recuperar	Permite visualizar documentos (não gerenciar) e analisar os dados não gráficos associados a eles.
Cadmandu	Pesquisar recuperar	Gerenciador de arquivos de um servidor. Permite checkout do servidor para revisão local. Permite visualizar um grande número de formatos.
AIM	negociar comunicar recuperar	AIM, constitui uma solução completa para o gerenciamento de informações. Oferece uma infra-estrutura aberta e orientada por objetos. Incluindo um visualizador de múltiplos formatos, um workflow gráfico e acesso a Internet.

Tabela 11 - Lista de produtos candidatos

## 5. CONCLUSÕES E PERSPECTIVAS FUTURAS

### 5.1 Conclusões

Este trabalho alcançou seu objetivo ao estabelecer um método de engenharia de software para o desenvolvimento de sistemas computacionais de apoio a projetos de engenharia (SCAPE). O método permite que o engenheiro de software desenvolva, com sucesso, um SCAPE para aplicações complexas seguindo as etapas aqui apresentadas.

A robustez do método pode ser comprovada pelos exemplos, onde não houve necessidade de um grande detalhamento do modelo para possibilitar a sua implementação. Esta robustez do método permite que ele possa ser usado por um engenheiro de software sem muito conhecimento da área de especialização do projeto, condição essencial ao seu bom desempenho.

O trabalho foi inédito ao estender o escopo de aplicação dos métodos de engenharia de software. Normalmente utilizados para desenvolver um programa de computador, este trabalho comprovou que os métodos de engenharia de software podem também ser usados, com sucesso, para desenvolver sistemas formados por produtos de software.

Com respeito à construção dos modelos dos SCAPE pode-se observar que:

- a) A adoção do diagrama de domínios como primeira abordagem de análise favorece a compreensão do problema, pois esta visão de alto nível orienta a elaboração dos diagramas de classes, que é realizada a seguir.
- b) O paradigma de objetos mostrou-se adequado para a representação dos SCAPE, promovendo uma perfeita sintonia entre os componentes dos sistemas analisados e suas representações no modelo. Os diagramas de classe oferecem uma excelente visão de conjunto do SCAPE, permitindo ao engenheiro de software produzir sistemas de elevada complexidade de um modo simples.
- c) A granularidade dos diagramas no domínio do produto deve ser o suficiente para se identificar todos os elementos individuais a serem projetados, e que estes possam ser destinados a uma equipe de trabalho específica.
- d) Os domínios do gerenciamento e de infra-estrutura são relativamente independentes das áreas de especialização imposta pelo produto. Esta constatação vem da observação de que os produtos de software usados para implementá-los podem ser aplicados em diferentes especialidades de SCAPE. Este fato sugere a possibilidade da existência de uma solução genérica para estes domínios, e a possibilidade de aumentar o nível de padronização nestas atividades.

- e) A característica cíclica dos processos de desenvolvimento e gerenciamento, caracterizada pela alternância das atividades do tipo modelar/decidir ou planejar/controlar, respectivamente, pode ser facilmente observada nos diagramas de eventos. Estes ciclos podem servir para garantir a precisão e completeza do modelo.
- f) O domínio da implementação, onde residem os produtos de software, ocupa uma posição de servidor dos demais domínios do SCAPE, mostrando a sua posição real situação de meio viabilizador da automação, e não de substituto do projetista.

Sobre a técnica de reutilização de produtos de software utilizada pode-se concluir que:

- a) A técnica de reutilização de software utilizada foi suficiente para encontrar uma solução viável nos casos estudados, no entanto, observa-se que as soluções possíveis para um SCAPE não são únicas. A falta de critérios adicionais para qualificar as soluções impede que se caracterize a solução mais adequada, há necessidade de se definir critérios adicionais de projeto para selecionar uma solução dentre as encontradas.
- b) O conjunto de soluções possíveis para um SCAPE, com a utilização deste método é condicionada pela qualidade e diversidade do banco de dados de produtos de software. A criação e manutenção deste banco é a etapa mais complexa

e trabalhosa da aplicação do método. Quanto mais abrangente for o banco de dados, e quanto mais precisa for a catalogação dos produtos, maior a chance de se encontrar um conjunto de produtos de software adequado para cada classe do modelo.

c) Não é possível se aplicar o método sem o auxílio de um suporte computacional adequado. Como resultado deste trabalho foi desenvolvido o programa BSCAPE que deve ser usado em conjunto com um programa CASE que adote o paradigma de objetos. Este conjunto de programas oferece o apoio computacional necessário para a aplicação do método em casos reais, e permite que uma pequena equipe de projeto desenvolva integralmente um SCAPE.

## **5.2 Perspectivas Futuras**

As perspectivas de trabalhos futuros nesta área estão associadas com a melhoria do método, a reutilização dos modelos, a melhoria dos programas de computador que suportam o método, uma maior abrangência do banco de dados de produtos de software e a otimização dos critérios de seleção de produtos.

A aplicação deste método em uma série de problemas sugere a existência de padrões comuns entre os SCAPE, que seriam beneficiados por uma solução única geral. Observou-se a relativa independência dos domínios do gerenciamento e da infra-estrutura com a áreas de desenvolvimento dos

produtos. Ao se criar uma solução única que possa ser adaptada a qualquer SCAPE nestes domínios, reduz-se o escopo de projeto apenas para o domínio onde a especialização é mais dependente da aplicação.

A contínua aplicação deste método em casos reais poderá levar à adoção do questionário de apoio à análise como um critério de avaliação da automação de projetos de um sistema. Hoje este questionário é um mero apoio à análise de um problema. O revisão do questionário neste sentido é uma nova área e aplicação surgida com o trabalho.

Para aumentar a precisão do método na busca de uma solução, deve-se aumentar a abrangência do banco de produtos de software. A solução ideal para este problema é descentralizar a atualização deste banco entre as próprias empresas fornecedoras dos produtos. A criação de um banco central ligado à rede Internet, permite que cada empresa criadora do produto atualize o banco de dados sobre eles a cada nova versão, ou nova funcionalidade adicionada. O banco de produtos de software deve ser desenvolvido por uma entidade que garanta a generalidade ao banco, baseando-se nas funções padronizadas pela arquitetura geral de um SCAPE aqui apresentada.

Um nova linha de pesquisa aberta com este trabalho encontra-se na busca pela otimização dos critérios para seleção produtos. É possível aumentar a automação no

processo de criação dos SCAPE, e aumentar a precisão das soluções encontradas, pela incorporação de novas técnicas de busca de produtos. O uso de técnicas de sistemas especialistas nos métodos de procura podem utilizar os elementos não estruturados da descrição do produto para facilitar a sua identificação, e levar a novas soluções.

Observa-se uma tendência internacional na indústria de software a uma maior "componentização" dos produtos. O método aqui apresentado pode ser estendido para qualquer classe de problemas que venha a ser resolvido pela integração de componentes de software. Para atingir-se este objetivo deve-se criar uma arquitetura geral para o tipo de problema a ser resolvido, que permita orientar a modelagem ao mesmo tempo classificar os componentes do banco. A generalização deste método de integração é uma promissora perspectiva futura para este trabalho.

## **Anexo A. DESCRIÇÃO DO BANCO DE DADOS DE PRODUTOS DE SOFTWARE**

### **1. Introdução**

Este anexo descreve o banco de dados de produtos de software que foi desenvolvido para permitir o cadastramento, localização e seleção dos produtos durante a construção de um SCAPE.

A seguir encontra-se a classificação usada para os programas, um diagrama de classes do banco de dados e uma amostra das interfaces dos programas desenvolvidos.

### **2. Classificação das Ferramentas de Software**

O Banco de Produtos de Software classifica cada programa segundo 4 critérios:

- Função,
- Plataforma de Processamento,
- Área de Aplicação e
- Padrões seguidos.

#### **a) Função**

A função que o software pode desempenhar no sistema, baseando-se nas funções padronizadas pela arquitetura geral de um SCAPE (item 2.3.3). Para aumentar o grau de detalhe

inclui-se nas funções modelar e decidir os métodos seguidos pelo produto para realizar esta função. Assim a tabela de funções fica:

Função	Descrição
planejar	dividir o trabalho em atividades.
delegar	distribuir atividades entre equipes de projeto.
controlar	observar (prazo, custo e requisitos ) sobre a execução das atividades.
realimentar	agir corrigindo desvios no planejamento, remuneração.
pesquisar	procurar dados e informações para realizar o projeto.
modelar	criar modelos sobre o produto em projeto, acrescentar novas informações.
modelar matemática	modelar com métodos matemáticos.
modelar gráfico	modelar com métodos gráficos (CAD).
modelar FEA	modelar pelo método dos elementos finitos (ou análogo).
modelar simulação	modelar para simulação.
modelar estatístico	modelar com métodos estatísticos.
modelar requisitos	modelar organizando requisitos (banco de dados).
decidir	analisar soluções e optar por uma solução proposta.
decidir otimizar	decidir com métodos e critérios de otimização.
decidir análise	decidir com base em resultado de análise.
decidir simular	decidir com base em simulações de comportamento.
negociar	influenciar nas decisões de projeto de outras equipes, compromissos.
arquivar	armazenar as informações de projeto, acompanhar a evolução.
comunicar	compartilhar decisões de projeto com demais membros das equipes.
recuperar	resgatar informações arquivadas, acompanhar projeto.
capturar	entrada de informações de projeto, mudança de mídia.
imprimir	saída de informações de projeto, mudança de mídia.
traduzir	conversão de formato entre informações de projeto.
facilitar	funções de apoio ao projeto, viabilizam as outras funções.
treinar	garantir a continuidade da evolução do sistema.

Tabela A-1 - Lista de funções dos produtos de software.

#### b) Plataforma de Processamento

Classifica os produtos quanto as famílias de processadores e sistemas operacionais nos quais o produto de software

pode ser executado. Busca-se garantir a possibilidade de co-existência dos produtos em uma mesma plataforma, viabilizando a troca de informações.

SIGLA	SIGNIFICADO
INTEL	Linha de Processadores INTEL 80X86
UNIX	Sistema Operacional UNIX ou similar
RISC	Linha de Processadores RISC em geral
MAC	Família de computadores Macintosh da Apple
SGI	Linha de computadores Silicon Graphics
DEC	Linha de Computadores Digital
IBM	Linha de computadores IBM
WIN	Sistema Operacional Windows 3.11, NT e 95
DOS	Sistema Operacional DOS
SUN	Linha de computadores SUN
HP	Linha de computadores HP

Tabela A-2 - Lista de plataformas utilizadas

### c) Área de Aplicação

Áreas de especialização onde o software pode ser aplicado. Segue-se, nesta classificação, uma das taxonomias descritas em GLASS; VESSEY (1995), e adaptada aos sistemas de apoio a projetos de engenharia, valorizando as áreas de interesse destes sistemas. Elas se encontram listadas na tabela abaixo:

Área de Aplicação
Aplicação Geral
Administrativa
Gerencial
Negócios
Financeira
Marketing
Transportes
Recursos Humanos
Ciências
Matemática
Médica

Social
Engenharia
Aeroespacial
Arquitetura
Nuclear e Energia
Civil
Mecânica
Elétrica
Naval
Nuclear
Construção
Química
Metalurgia
Manufatura
Computação
Controle
Automação
Militar
Publicação
Tempo Real
Linguagem de Programação
Sistema Operacional
Sistema de Rede
IHM
Simulação
Biblioteca de Programas
Banco de Dados
CASE
CASE Objetos
CASE Estruturado
CASE Requisitos

Tabela A-3 - Lista de áreas de aplicação

**d) Padrões seguidos**

Classifica os produtos segundo os padrões disponíveis no produto, para troca de informação de projeto. Inclui-se os formatos dos arquivos suportados pelo software, linguagens de programação disponíveis e outras formas de integração da informação.

SIGLA	SIGNIFICADO
TXT	Formato Texto, padrão ASCII
DOC	Formatos de Editor de Texto
BMP	Formato de figura, BitMap
CAL	Formato de figura, padrão CALS
HPGL	Formato de Plotagem

SGML	Formato de texto, padrão CALS
DXF	Formato de arquivo gráfico
IGES	Formato de arquivo gráfico
STEP	Formato para troca de informações técnicas
OLE	Padrão de Documento Composto do COM
CORBA	Padrão para documento composto do SOM
HTML	Padrão de acesso à Internet
SQL	Padrão de Linguagem para Banco de Dados
VBA	Visual Basic for Applications (linguagem)
JAVA	Linguagem JAVA para customização

*Tabela A-4 Lista de formatos de arquivos considerados no banco de produtos de software*

### **3. Construção do BDSOFT**

Para permitir a catalogação dos produtos de software, segundo os critérios de classificação acima, foi construído no sistema gerenciador de banco de dados ACCESS, o Banco de Produtos de Software - BDSOFT. O programa BDSOFT foi projetado usando o método OMT de modelagem orientado a objetos, e implementado na linguagem Visual BASIC no ambiente Windows.

#### **a) Modelo de objetos do BDSOFT**

A figura abaixo mostra o diagrama de classes do banco de produtos de software. Nele são identificados os atributos e as operações implementadas no banco para o cadastro de empresas e seus produtos, e a especificação das interfaces que operam o banco.

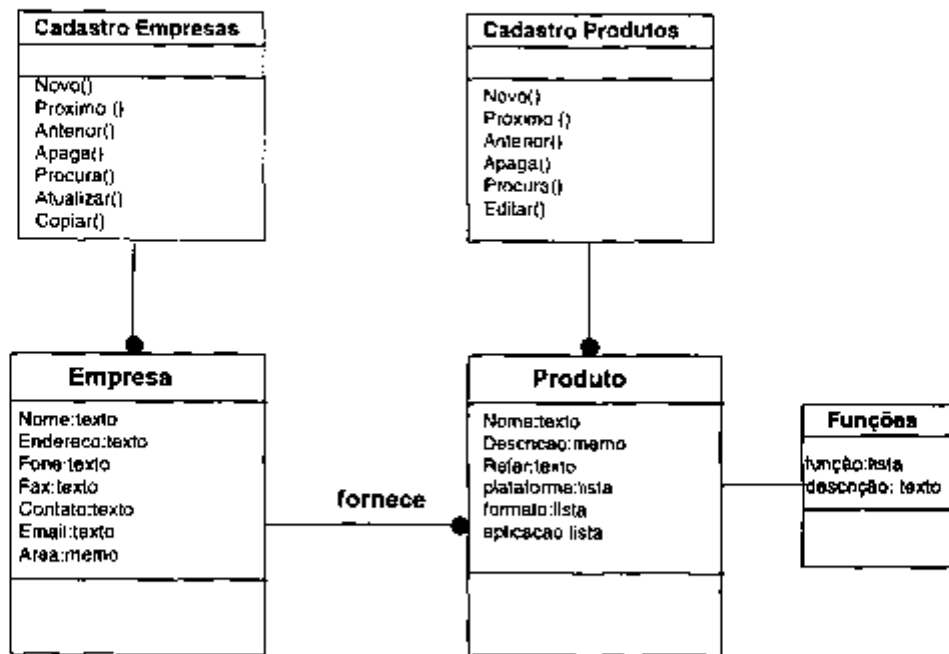


Figura A-1- Modelo de objetos do Banco de Produtos de Software

#### b) Implementação do BDSOFT

A implementação do BDSOFT resulta nas interfaces abaixo para o sistema operacional Windows 95. Detalhes da operação dos programas podem ser obtidos na ajuda que está disponível durante o processamento dos programas.

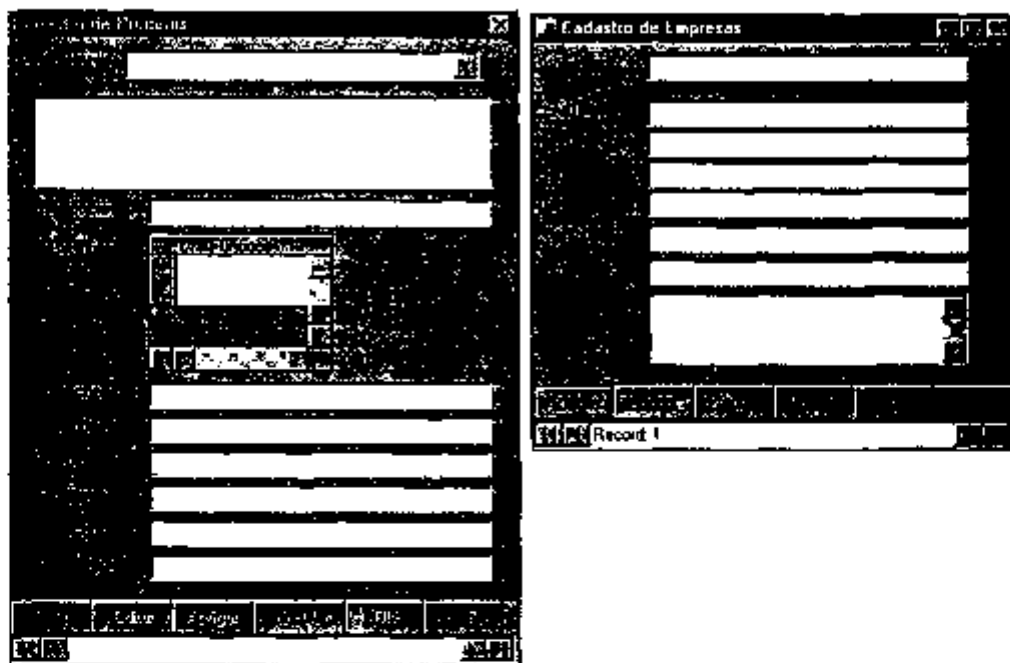


Figura A-2 - Interface de cadastro de empresas e produtos.

## **Anexo B. INTRODUÇÃO À MODELAGEM ORIENTADA POR OBJETOS**

Este resumo, baseado em CADRE (1993), faz uma introdução à notação utilizada na construção dos modelos orientados por objetos. Esta notação é extraída do método OMT - Object Modeling Technique de RUMBAUGH et al. (1993), acrescido da notação do diagrama de domínios, como proposto por SHLAER; MELLOR (1993).

### **1. Conceitos Fundamentais**

A modelagem orientada por objetos é um meio para se descrever os sistemas do mundo real. Um objeto representa uma abstração de uma entidade do mundo real, combinando, em um mesmo elemento informação e comportamento. Outros paradigmas usados no desenvolvimento de software (funcional ou dados) ligam apenas fracamente a informação e o comportamento. A orientação a objetos conduz a uma união mais bem definida destes dois conceitos.

COAD; YOURDON (1992) observam que o ser humano emprega normalmente três métodos para analisar o mundo real: a diferenciação, quando identifica os diferentes objetos do mundo real; a distinção entre o todo e suas partes, quando decompõe um objeto em suas partes componentes, e a

classificação, quando agrupa objetos assemelhados segundo algum aspecto comum. A análise baseada em objetos se utiliza destes mesmos métodos para desenvolver seus modelos.

São características importantes para a análise baseada em objetos:

- **Abstração**, que é o princípio de se ignorar os aspectos de um assunto não relevantes à um determinado propósito. A abstração é importante para entender um problema quando é impossível se considerar toda a sua complexidade.
- **Identidade** que garante que todos os objetos sejam distinguíveis através de um identificador. Apesar de todos os atributos de dois objetos poderem ser os mesmos, a identidade permite distinguí-los.
- **Classificação** que é a habilidade de se agrupar em classes, os conjuntos de objetos com a mesma estrutura de dados e comportamento.
- **Herança**, que está associada à classificação, é o princípio onde as características de uma classe são compartilhadas pelos elementos dela. A herança permite ao analista expressar características de uma classe uma única vez, e compartilhá-las com seus membros, simplesmente identificando-os como membros desta.

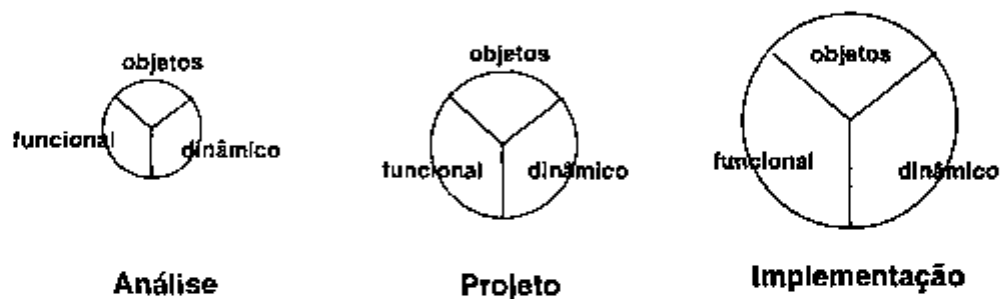
- **Polimorfismo** que é a habilidade de se ter uma mesma operação com comportamentos diferentes para diferentes classes. A implementação de uma resposta à uma operação é parte do próprio objeto, deste modo, há uma hierarquia de funções assim como há uma hierarquia de dados.
- **Encapsulamento**, ou ocultação de informações, que permite a separação dos aspectos internos dos objetos dos aspectos acessíveis por outros objetos. Esta característica impede que o sistema se torne excessivamente interdependente, evitando aumentar a sua complexidade e facilitando sua manutenção.

## **2.0 método OMT**

Uma metodologia de engenharia de software é um processo para a produção organizada de software, usando uma coleção de técnicas e convenções de notação. O ciclo de vida de um software cobre as fases de formulação do problema, análise, projeto, implementação, testes de operação e manutenção.

O método de modelagem orientada por objetos OMT, foi proposto por RUMBAUGH et al. em 1993. Ele consiste em se construir um modelo de objetos do domínio da aplicação e se adicionar, gradativamente, à ele detalhes do domínio do sistema de software, de modo que o mesmo modelo possa evoluir da fase de análise, para o projeto e daí para a implementação.

O sistema, no método OMT, é representado por um modelo representado por um conjunto de diagramas: modelo de objetos, modelo dinâmico e modelo funcional. Cada modelo representa uma visão ortogonal do sistema em projeto, que se complementam para formar o sistema final. A mesma notação é usada durante todo o ciclo de desenvolvimento. A figura representa a evolução do projeto de sistemas através do crescimento de detalhes dos modelos.



*Figura B-1 Evolução do sistema no método OMT*

Cada modelo é descrito por um conjunto de diagramas, que forma o suporte gráfico para representar o sistema. A tabela abaixo mostra os diagramas usados no método OMT:

<b>Modelo</b>	<b>Diagramas usados</b>
Modelo de objetos	diagrama de classes
Modelo dinâmico	diagrama de estados diagrama de eventos
Modelo funcional	diagrama de fluxo de dados

### **3. Notação usada nos diagrama**

No método de integração de SCAPE, usaremos a mesma notação dos diagramas do método OMT, acrescido do diagrama de

domínios. A notação e o significado dos diagramas são apresentados a seguir.

#### **a) Diagrama de Domínios**

O diagrama de domínio é uma ferramenta auxiliar na criação de um modelo de um sistema de software. O diagrama de domínio apresentado aqui segue os princípios propostos por SHLAER; MELLOR (1993), na descrição do método de desenvolvimento de software que leva os seus nomes. A divisão em domínios constitui o primeiro nível de particionamento do sistema. Um sistema complexo pode ser dividido em conjuntos distintos e independentes chamados de domínios. Cada domínio enfoca uma parcela de responsabilidade do sistema. Partindo de um enfoque próximo ao cliente, passa para abordagens de projeto até a implementação final. Os assuntos são tratados internamente aos domínios separadamente dos demais. Nele são desenvolvidos modelos próprios, estabelecendo relações entre os objetos, estados e funções que constituem cada domínio.

Problemas semelhantes tendem a possuir diagrama de domínio semelhantes. Cada domínio, imaginado como independente, pode ser também um componente reutilizável, para outros sistemas.

Os domínios são representados graficamente, conforme SHLAER; MELLOR (1993), por ovais, interligados por flechas duplas; que representam relações cliente-servidor existentes entre eles. A ponta da flecha indica o domínio servidor, e a origem o domínio cliente, como mostra a figura abaixo:

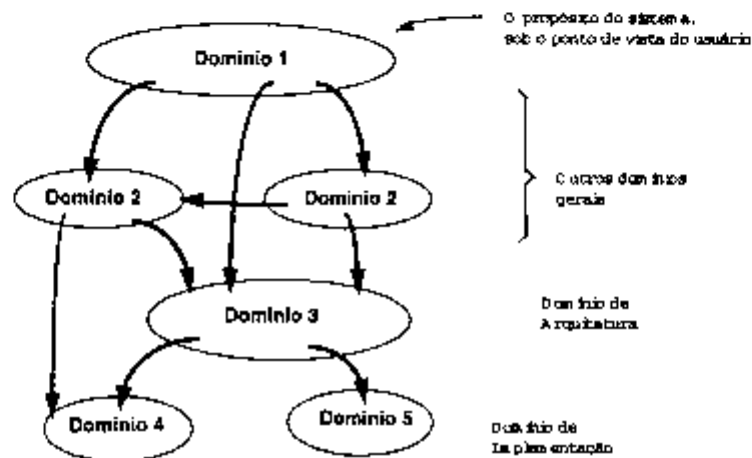


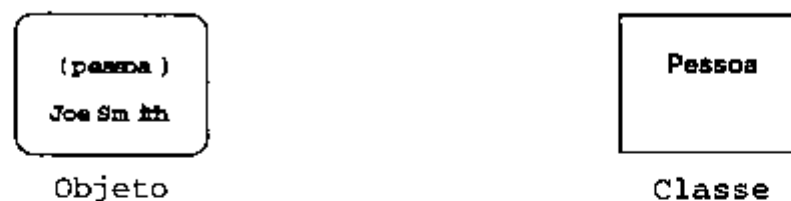
Figura B-2 - Exemplo de um diagrama de domínio (SHLAER; MELLOR, 1993)

O domínio que age como cliente conta com os serviços do domínio servidor para a execução de certas operações. Esta relação é semelhante à transição feita da fase de análise do projeto de sistemas, realizada no domínio da aplicação, para a fase de implementação. A análise, enquanto modelo da aplicação, é cliente dos serviços a serem construídos durante a implementação. Existe assim uma relação entre o particionamento de um sistema em domínios e as fases do seu desenvolvimento. Os domínios clientes correspondem as fases anteriores do desenvolvimento, enquanto que os domínios servidores estão associados às fases posteriores.

O método de integração de SCAPE usa os princípios de domínio para estabelecer, já no início da modelagem, uma arquitetura comum ao SCAPE, e orientar o seu desenvolvimento.

#### **b) O Diagrama de classes**

O propósito dos objetos é representar o que é realmente relevante de cada domínio. Os objetos permitem o entendimento do mundo real, pela representação de seus elementos, e visam fornecer uma base para a implementação da solução. Usa-se instância de objeto a um representante preciso do conjunto de objetos, e chama-se de classe de objetos a um grupo de elementos semelhantes, que compartilham comportamentos, semântica e relacionamentos com outros objetos.



*Figura B-3 Representação gráfica de uma classe e um objeto*

Os atributos guardam os valores dos dados de um objeto. Para cada instância de um objeto o atributo possui um valor. As operações são transformações aplicadas ao objeto, ou por estes a uma classe. Os objetos de uma classe compartilham as operações dela. Uma mesma operação, quando aplicada à diferentes classes, é dita polimórfica, isto é,

toma formas diferentes em classes diferentes. Um método é a implementação de uma operação para uma classe.

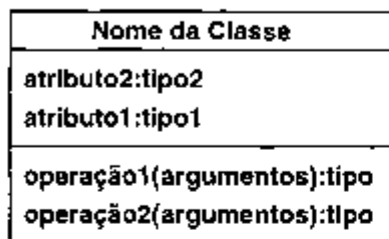


Figura B-4 Representação de uma classe

Além dos objetos o modelo é composto por relações entre os objetos e classes. Estas relações podem ser associações, generalizações e agregações. Cada tipo de relação possui um significado diferente no contexto do modelo, e refere-se a uma relação existente no mundo real.

As relações entre classes são chamadas *associações*. Uma associação descreve um grupo de ligações entre objetos, que compartilham uma estrutura e uma semântica. O número das instâncias de uma classe, que se relacionam com outra classe, é chamada de *multiplicidade*, como mostra a figura abaixo.

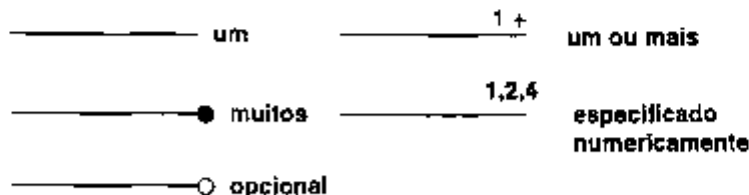


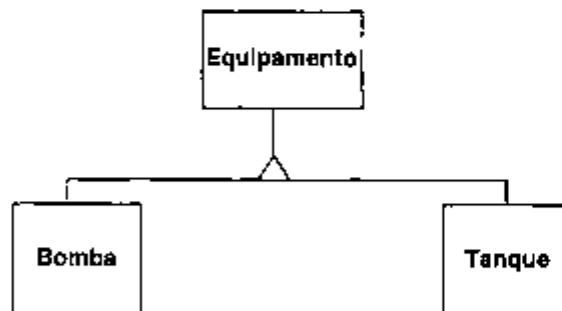
Figura B-5 Representação da multiplicidade em uma associação.

Uma ligação também pode possuir atributos. Na notação utilizada, um atributo de ligação é representado por um quadro ligado à associação.



*Figura B-6 Exemplo de atributo de ligação*

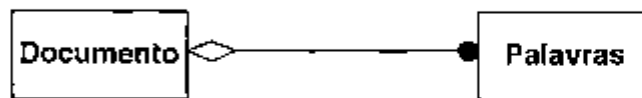
A ligação do tipo generalização ou herança estabelece a relação entre uma classe e as versões refinadas dela. A classe a ser refinada é chamada de *superclasse*, e cada versão, *subclasse*. A subclasse herda as características (atributos e operações) da superclasse.



*Figura B-7 Exemplo de estrutura de herança*

A ligação de agregação é o relacionamento do tipo todo-parte, onde os objetos representam componentes de um outro objeto. É representada por uma linha com um losango junto à extremidade estrutural (todo). A agregação é um conceito especial de associação, estabelecendo uma relação de

composição. É possível existir, inclusive, a agregação recursiva.



*Figura B-8 Exemplo de agregação*

Existem elementos do mundo real que descrevem outros elementos também do mundo real. Considerando um objeto como padrão, ele transfere os atributos comuns aos elementos deste padrão. Os elementos individuais são instâncias deste padrão. A figura abaixo indica que o objeto ModeloDeCarro é um padrão, ou metadados, do objeto carro. Esta é a notação usada para a instanciação e para o relacionamento entre uma metaclassa (padrão) e uma classe (instância).



*Figura B-9 - Exemplo da notação de padrões*

### **c) O diagrama de eventos**

Após o entendimento do sistema sob o ponto de vista estático, representado pelo modelo de objetos, passa-se a examinar as mudanças ocorridas com o sistema no decorrer do tempo. Desenvolve-se para isto o modelo dinâmico, composto pelos eventos e pelos estados do sistema.

Um evento corresponde à transmissão de informação unidirecional de um objeto para outro. Um estado é uma abstração de um objeto, definido pelo conjunto de atributos e suas ligações. Com o tempo, os objetos estimulam-se uns aos outros, resultando em uma série de alterações em seus estados. O estímulo individual de um objeto para outro é um evento. A reação de um objeto a este estímulo pode ser a alteração do seu estado, e, eventualmente, a geração de outro evento.

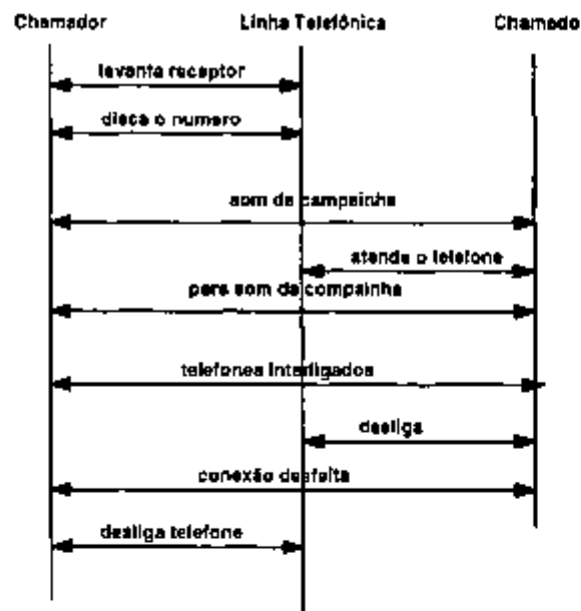


Figura B-10 Diagrama de eventos para uma chamada telefônica

O diagrama de evento, apresentado acima, representa os eventos entre os objetos do sistema.

#### d) O diagrama de estados

O diagrama de estados, mostrado abaixo, representa os estados e eventos ocorridos em uma única classe. Nota-se que um evento separa dois estados, e um estado separa dois eventos.

Um diagrama de estados descreve o comportamento de uma única classe de objetos, e pode representar simples ciclos de vida, ou laços contínuos. Cada objeto é independente de outros objetos e comporta-se a seu próprio modo. O modelo dinâmico é uma coleção de diagramas de estados que interage uns com os outros por intermédio de eventos compartilhados.

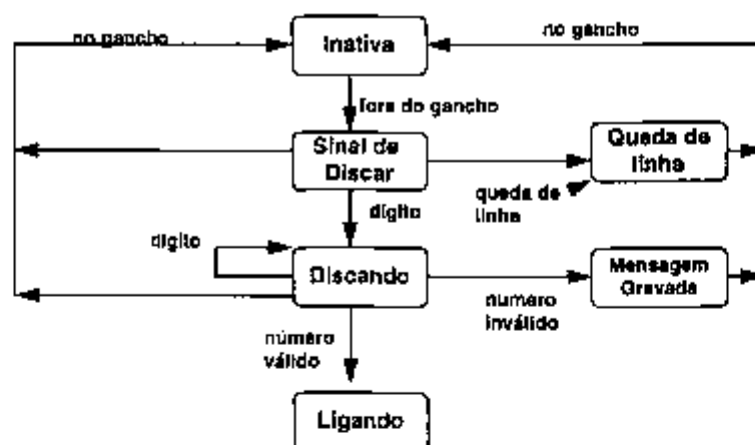


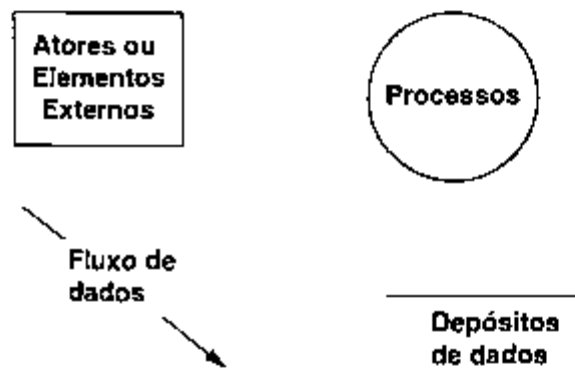
Figura B-11 Exemplo de diagrama de estados como um laço contínuo.

#### e) O diagrama de fluxo de dados

O modelo funcional especifica os resultados de um processamento sem especificar como ou quando eles serão processados. Ele especifica o significado das operações do

modelo de objetos e as ações do modelo dinâmico. Este modelo descreve como os dados são transformados, mas não quem ou quando isto ocorrerá.

O modelo funcional é composto por múltiplos diagramas de fluxos de dados (DFD), que especificam o significado das operações e restrições. Um DFD contém processos que transformam dados, fluxos de dados que movimentam os dados, objetos atores que produzem e consomem dados, e objetos depósitos, que armazenam dados passivamente.



*Figura B-12 Notação de Yourdon para os Diagramas de Fluxo de Dados*

A figura abaixo exemplifica um DFD para controle de uma conta corrente bancária:

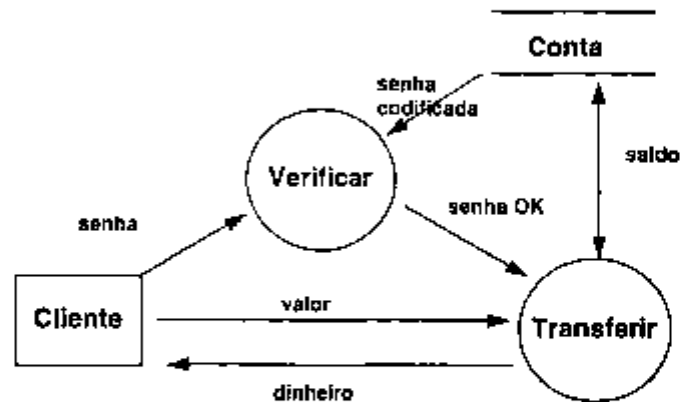


Figura B-13 DFD para uma movimentação em Contracorrente

#### 4. Conclusão

O desenvolvimento de sistemas baseado em objetos pode ser empregado com sucesso na grande maioria das aplicações. Pode-se esperar benefícios com um melhor gerenciamento da complexidade dos sistemas, e no aumento da produtividade dos programadores, como consequência do encapsulamento e da reutilização dos códigos. A herança permite que se adicione, ou elimine objetos sem comprometer a aplicação como um todo aumentando a capacidade de manutenção, assim como a capacidade de expansão dos sistemas, reduzindo também os custos de longo prazo.

Os modelos desenvolvidos pelo método OMT são intimamente relacionados entre si. O modelo funcional mostra o que um sistema deve fazer, detalhando os processo como operações sobre os objetos. Cada processo é implementado como um método sobre algum objeto. O modelo dinâmico mostra as seqüências em que as operações são executadas. Os atores

são objetos explícitos no modelo de objetos. Os fluxos de dados representam operações sobre os objetos ou por eles executadas. O modelo dinâmico para um objeto ator especifica quando ele atua. Os depósitos de dados são também objetos, ou pelo menos fragmentos de objetos, representados como atributos. Os fluxos de dados são valores no modelo de objetos.

A construção de bibliotecas de classes facilita a reutilização de classes, objetos e código. A representação de classes ricas em conteúdo e relacionadas com elementos do mundo real, torna a programação mais intuitiva e facilita a comunicação entre cliente e projetista.

## **Anexo C. QUESTIONÁRIO DE APOIO À ANÁLISE DE UM SCAPE**

### **1. Introdução**

Este questionário objetiva apoiar o engenheiro de sistemas na análise de um sistema computacional de apoio a projetos de engenharia já existente, e avaliar seu estágio de automação. As questões apresentadas foram adaptadas de um questionário semelhante proposto por CARTER; BAKER (1991).

O desenvolvimento de técnicas para avaliação de processos de engenharia, tem se intensificado com o crescimento da preocupação com a qualidade, e a necessidade de levantar dados para um controle estatístico destes processos. A indústria de software, através do Software Engineering Institute, criou um modelo de maturidade de 5 estágios e um critério de avaliação do processo de produção de software, como apresenta HUMPHREY (1988) e ZUBROW et al. (1994). Outras especialidades, como a mecânica e eletrônica, possuem questionários próprios de avaliação, como os citados por CARTER; BAKER (1991).

### **2. Construção do Questionário**

O presente questionário não deseja criar um modelo de maturidade, mas sim fornecer subsídios para que o

engenheiro de software possa adotar uma abordagem mais formal nos primeiros passos da modelagem do sistema.

O questionário é construído sobre as funções existentes no modelo geral de um sistema de projetos, descrito no Capítulo 2. Para cada função são feitas quatro perguntas, algumas delas associadas às quatro fases na evolução da automação, ou seja

- 1a questão: analisa-se a existência, ou não daquela função no SCAPE;
- 2a questão: analisa-se a existência de algum tipo de automação desta função no SCAPE;
- 3a questão: analisa-se a abrangência da automação, e a integração desta por toda a organização.
- 4a questão: analisa-se a extensão da empresa virtual, e a integração entre clientes, projetistas e fornecedores.

### **3. Instruções para se responder o questionário**

Para a avaliação poder ser a mais confiável possível, deve-se responder apenas com as atuais práticas de trabalho, evitando considerar um ambiente futuro, com melhorias ou propostas que ainda estão em fase de implantação. As respostas devem-se concentrar em um único produto, ou em produtos que compartilhem os mesmos recursos para projeto.

Cada questão é respondida com sim, não ou não se aplica, havendo possibilidade de tecer comentários sobre a

resposta. Elas são reunidas em grupos de 4, por assunto, em uma seqüência crescente de complexidade. As respostas são transcritas para a folha de resposta, onde pode-se realizar uma análise visual do perfil da automação de projetos neste sistema.

## Questionário de Avaliação

Companhia: \_\_\_\_\_ Data : \_\_\_\_\_

Responsável pelas respostas : \_\_\_\_\_

Produto/ Linha de produto analisado : \_\_\_\_\_

### Planejar

**Definição:** Dividir o trabalho em atividades, com base nos requisitos do produto. Procurar uma divisão que aumente a independência das atividades, permitindo a simultaneidade do trabalho.

**Avalia-se:** A existência de planejamento automatizado com capacidade para analisar os caminhos críticos e cenários alternativos.

1. Existe um planejamento de curto prazo, realizado pelos projetistas anterior à realização de uma tarefa? Comente sobre a eficiência do planejamento.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

2. A organização possui um planejamento automatizado que permite identificar o caminho crítico e analisar cenários alternativos? Comente sobre as ferramentas de planejamento disponíveis.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

3. A organização dispõe de um planejamento estratégico pluri-anual para cada família de produtos?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

4. O planejamento inclui a participação dos clientes e fornecedores?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### Delegar

**Definição:** criar equipes multidisciplinares de projeto, e distribuir grupos de atividades entre elas. Eventos caracterizam os grupos de atividades.

**Avalia-se:** divulgação das atividades, criação das equipes e delegação de responsabilidade

5. Os detalhes das tarefas e as prioridades de cada projetista são devidamente comunicados e compreendidos ?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

6. O processo de desenvolvimento do produto está difundido por entre as equipes de trabalho multidisciplinares?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

7. Os requisitos, especificações, interdependências e prioridades do produto são discutidos por toda a equipe de projeto?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

8. Os requisitos, especificações, interdependências e prioridades do produto são discutidos com clientes e fornecedores?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### Controlar

**Definição:** Observar a execução das atividades no tocante a prazo, custo e atendimento aos requisitos.

**Avalia-se:** Existência de um sistema de informação gerencial para controle do processo de projeto.

9. O planejamento é medido com índices objetivos ? Comente os índices utilizados e sua capacidade de medição.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

10. Os índices são registrados em um sistema de informações e as correções necessárias são analisadas?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

11. O sistema de informação está integrado aos demais sistemas da organização?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

12. Os clientes e fornecedores participam da análise e correção do planejamento?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

### **Realimentar**

**Definição:** Atuação no sentido de corrigir desvios dos requisitos do produto, e incentivar os bons resultados.

**Avalia-se:** Existência de um sistema de informação gerencial para controle do produto. (ex. Controle de configuração)

13. Os problemas encontrados nos produtos são registrados e analisados ? Comente alguns problemas encontrados nos produtos.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

14. Os problemas registrados em um sistema de informação que prioriza, programa e acompanha a correção?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

15. O sistema de informação é utilizado como banco de apoio à decisão e como indicadores da satisfação do cliente?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

16. A tendência dos itens de controle, relatórios de problemas e pedidos de melhorias são analisados visando a melhoria contínua do produto?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### **Pesquisar**

**Definição:** Procura por informações e dados necessários ao trabalho de uma equipe de projeto.

**Avalia-se:** Uso de sistemas computacionais na representação e recuperação dos dados dos produtos. (ex. CAD) Integração entre os sistemas.

17. Existem ferramentas computacionais para apoiar as atividades de projeto? Comente sobre as ferramentas de modelagem disponíveis.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

18. As ferramentas de construção de modelos são utilizadas por todos os membros da equipe para troca de dados?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

19. As ferramentas de projeto estão integradas de modo a se influenciarem mutuamente?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

20. As ferramentas estão integradas por todo o empreendimento, incluindo clientes e fornecedores?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### **Calcular**

**Definição:** Acrescentar informações aos modelos do produto, criando soluções para os problemas de projeto, ou analisando soluções propostas.

**Avalia-se:** Existência de análise automatizada dos modelos do produto (ex. Sistemas CAE). Integração dos sistemas de representação e a análise. (CAD/CAM).

21. As decisões de projeto são documentadas, metodicamente, de modo que os projetos sejam consistentes e repetíveis?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

22. Existe alguma metodologia de análise para avaliar e medir e determinar falhas no comportamento do produto durante a fase de projeto ? Comente sobre as ferramentas disponíveis.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

23. Existem métodos computacionais adequados à integração do modelo produto ao processo de produção?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

24. Métodos analíticos, possivelmente computacionais, são usados para avaliar: custo, testabilidade, compatibilidade, manufaturabilidade e suportabilidade do projeto, antecipando a satisfação do cliente?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### **Decidir**

**Definição:** Tomada as decisões de projeto, selecionando soluções no sentido de otimizar o atendimento aos requisitos do produto.

**Avalia-se:** A tomada de decisão no processo de projeto, o seu nível de automação e a participação de todos os envolvidos.

25. As decisões de projeto são tomadas pelos projetistas, supervisores ou gerentes, independentemente?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

26. As decisões de projeto são negociadas dentro de uma equipe multidisciplinar?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

27. São usadas ferramentas de suporte à decisão e gerenciamento do processo interdisciplinar? Comente sobre as ferramentas disponíveis.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

28. Todos os integrantes do projeto, incluindo os clientes e fornecedores, participam das decisões de projeto?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### **Negociar**

**Definição:** Influenciar nas decisões de projeto de outras equipes ou de outros membros da equipe.

**Avalia-se:** A existência de um repositório central dos modelos do produto, que auxilie nas negociações e decisões em grupo.

29. Existe um modelo eletrônico do projeto (ex. CAD), que representa o produto desde a sua fase de concepção? Comente sobre os formatos do modelo.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

30. O modelo de projeto procura ser único para cada disciplina de projeto? Comente sobre as disciplinas e seus modelos.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

31. Existe uma visão única do projeto, compartilhada por todas as equipes multidisciplinar e pela produção?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

32. Existe uma visão única do projeto, que é compartilhada pelos clientes e fornecedores?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### Arquivar

**Definição:** Manter as informações de projeto disponíveis às equipes de projeto, e permitir o acompanhamento da evolução do projeto, através delas.

**Avalia-se:** Existência de um banco de documentos do produto, disponível para o projeto.

33. Existem dados eletrônicos do projeto que devem ser arquivados? comente sobre o formato e quantidade destes dados.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

34. Os dados arquivados estão disponíveis a todos os projetistas de uma mesma equipe?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

35. Os documentos estão organizados de modo a minimizar a redundância, duplicidade e dubialidade de informações?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

36. Os dados estão organizados em um banco de dados, de modo a oferecer informações sobre o andamento do projeto?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### Comunicar

**Definição:** Compartilhar as decisões de projeto com os demais membros da equipe.

**Avalia-se:** Existência de uma rede de comunicação de dados, e abrangência da rede nas atividades de projeto. (ex. Intranet, Workflow, Internet e EDI)

37. O sistema de projeto dispõe de uma rede de comunicação de dados? Comente sobre a abrangência e a qualidade das comunicações da rede.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

38. A rede de comunicação de dados dispõe de um correio eletrônico para troca de mensagens e documentos entre os projetistas?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

39. O fluxo de documentos é gerenciado de modo a indicar o andamento do projeto, e eventuais mudanças de fluxo de documentos?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

40. A rede permite o fluxo externo de documentos , por exemplo com acesso à Internet? Ou um sistema de EDI técnico?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

### Recuperar

**Definição:** Resgatar dados e informações arquivadas, e disponibilizar estas informações aos projetistas.

**Avalia-se:** Disponibilidade de postos de trabalho, acessibilidade e qualidade das informações nos postos de trabalho.

41. Os projetistas dispõem de postos de trabalho, onde os dados dos produtos estão sob sua responsabilidade? Comente sobre a disponibilidade de postos de trabalho para os projetistas.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

42. Os componentes em cada disciplina de projeto tem acesso a todos os dados do produto relativos à sua disciplina? Comente sobre a recuperabilidade de documentos arquivados por outros projetistas.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

43. Os componentes em uma disciplina de projeto tem acesso eletrônico aos dados do produto, relativos às diferentes disciplinas envolvidas no desenvolvimento do produto, a dados de outros produtos?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

44. Os projetistas e as equipes tem acesso eletrônico aos dados corporativos dos produtos em desenvolvimento, incluindo os dados vindos de clientes e fornecedores, e projetos anteriores?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

### **Capturar**

**Definição:** Entrar informações de projeto no sistema, convertendo outros documentos em documentos eletrônicos.

**Avalia-se:** Uso de documentos eletrônicos, recursos de digitalização e a política de uso dos documentos eletrônicos.

45. O sistema de projeto faz uso de documentos eletrônicos? Comente sobre a abrangência e uso destes documentos

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

46. Os documentos convencionais podem ser convertidos em documentos eletrônicos, através de um processo padronizado?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

47. Este uma política de conversão, controle de qualidade dos documentos eletrônicos que é válida para toda a organização?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

48. Os documentos eletrônicos são trocados com clientes e fornecedores, e produção?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### Imprimir

**Definição:** Sair informações de projeto, convertendo documentos eletrônicos em outras mídias.

**Avalia-se:** Disponibilidade de recursos para geração de documentos convencionais, controle da emissão de documentos convencionais.

49. Existem recursos computacionais para impressão dos documentos eletrônicos?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

50. Existe um sistema padronizado de editoração eletrônica e apresentação? Comente sobre os recursos disponíveis.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

51. A saída de documentos é padronizada e controlada, de modo a evitar a concorrência entre documento eletrônico e o convencional?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

52. A saída de documentos da empresa é feita em um local centralizado, de modo a garantir a confiabilidade da informação?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### Traduzir

**Definição:** Conversão de formato entre documentos, permitindo a integração da informação.

**Avalia-se:** Extensão do uso de padrões de documentos no projeto, e eficiência da tradução e integração das informações.

53. Durante o desenvolvimento do produto as especificações e demais documentos de projeto são padronizados ? Comente sobre os formatos adotados, e sua abrangência no projeto.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

54. Os dados de uma equipe pode ser transferido para outra disciplina? Existe compatibilidade entre os formatos? Comentes sobre os programas de tradução utilizados.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

55. Os modelos do produto em desenvolvimento são compatíveis com diversas ferramentas de projeto? Comente sobre a compatibilidade entre as ferramentas disponíveis.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

56. Os padrões de formatos são compatíveis com os adotados com clientes e fornecedores?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### **Facilitar**

**Definição:** Operações que não estão diretamente ligadas ao projeto, mas viabilizam, simplificam ou agilizam as outras funções do sistema.

**Avalia-se:** Existência de um programa interno de qualidade, extensão da padronização nos produtos, componentização e reutilização.

57. A organização utiliza de padrões para garantir a qualidade do produto? Comente sobre as normas e padrões utilizados.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

58. Existe um programa interno de qualidade, preocupado em elaborar padrões para as equipe de projeto?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

59. Os padrões são usados para garantir a testabilidade, manufaturabilidade, suportabilidade, usabilidade e qualidade dos produtos?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

60. Os padrões estão de acordo com entidades certificadoras internacionais, e são revistos regularmente em melhora contínua?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

### Treinar

**Definição:** Garantir a continuidade da evolução do sistema de projeto, mantendo e aumentando as especializações dos projetistas.

**Avalia-se:** Qualificação da equipe de projeto e existência de um programa de educação continuada.

61. Os projetistas são qualificados proporcionalmente à complexidade do projeto?

Comente sobre a qualificação da equipe.

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

62. Os integrantes do projeto estão capacitados a utilizar as ferramentas disponíveis para o projeto?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

63. Existe um programa de treinamento para auxiliar o relacionamento em equipe, e que considere também os procedimentos, ferramentas e padrões a serem utilizados?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_

64. O treinamento se estende para toda a equipe envolvida no empreendimento, incluindo os clientes e fornecedores?

Sim       Não       Não se aplica

Comentários : \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Quadro de Respostas

Companhia: \_\_\_\_\_ Data : \_\_\_\_\_

Responsável pelas respostas \_\_\_\_\_

Produto/ Linha de produto analisado : \_\_\_\_\_

Anote apenas as respostas SIM.

Estágio de Automação:		inicial	médio	alto	muito alto
Gerenciamento	Planejamento	1	2	3	4
	Delegação	5	6	7	8
	Controle do	9	10	11	12
	Realimentação	13	14	15	16
Desenvolvimento	Resquisar por dados	17	18	19	20
	Concepção e	21	22	23	24
	Tomada de decisão	25	26	27	28
	Negociação	29	30	31	32
Infra-estrutura	Arquivamento	33	34	35	36
	Comunicação	37	38	39	40
	Recuperação	41	42	43	44
	Captura de Dados	45	46	47	48
	Impressão	49	50	51	52
	Tradução	53	54	55	56
	Padronização	57	58	59	60
	Treinamento	61	62	63	64

#### **4. Análise das respostas**

Transcritas as respostas para o quadro, pode-se realizar uma análise visual indicando pontos fortes (Sim) e fracos (Não).

Uma primeira análise é feita com relação aos três grandes grupos de respostas: gerenciamento, desenvolvimento e infra-estrutura. Deve haver um equilíbrio entre eles, sendo esperado um número proporcional de respostas "sim" em cada grupo.

A segunda parte da análise parte para itens específicos verificando as respostas negativas de cada grupo.

Caso o número de respostas afirmativas for baixo, deve-se procurar desenvolver um plano de automação, procurando aproximar o ambiente de desenvolvimento analisado ao modelo proposto. O questionário pode servir como auxílio na elaboração do plano de automação, indicando o caminho para aumentar o número de respostas positivas. A aplicação periódica do questionário pode servir também como uma medida indireta da evolução da automação no sistema.

#### **5. Programa ANALISE**

Como parte deste trabalho desenvolveu-se o programa de computador ANALISE para automatizar o processo de aplicação do questionário. Este programa, exemplificado pela figura

abaixo, permite arquivar as respostas e os comentários realizados, emitindo um relatório que resume a avaliação.

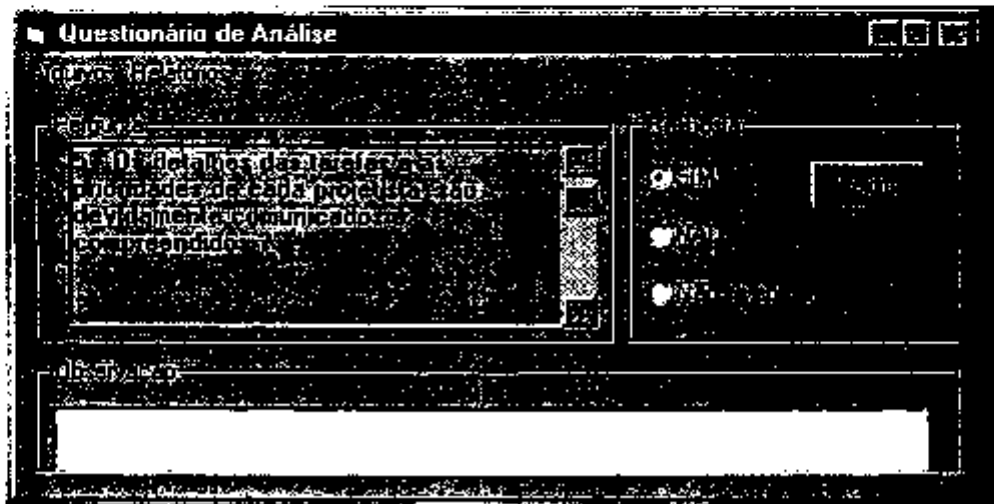


Figura C-1 - Interface do programa ANALISE

## **Anexo D.**

### **INSTRUÇÕES PARA USO DO PROGRAMA BSCAPE**

#### **1. Introdução**

Este anexo descreve o uso programa BSCAPE desenvolvido para dar apoio ao engenheiro de software na implementação dos SCAPE. O programa foi desenvolvido na linguagem Visual Basic 4.0, para o sistema operacional Windows 95. Detalhes maiores sobre a operação deste programa podem ser obtidos diretamente durante a operação do programa.

#### **2. Instalação do programa BSCAPE**

O programa BSCAPE é instalado executando-se o programa SETUP.EXE localizado no Disco de Instalação No.1. Os demais discos são solicitados automaticamente ao usuário. Ao final, a instalação cria um atalho para o programa BSCAPE no menu principal do Windows. Clique duas vezes neste atalho para executar o programa BSCAPE.

#### **3. Seqüência de passos para uso do BSCAPE**

##### **a) Criando o Banco de Produtos**

O primeiro passo para uso do programa é construir o Banco de Produtos de software que será consultado durante a implementação. Este banco encontra-se descrito no Anexo A,

e pode ser acionado no programa BSCAPE selecionando-se a opção "Banco de Produtos" do menu, como mostra a figura:

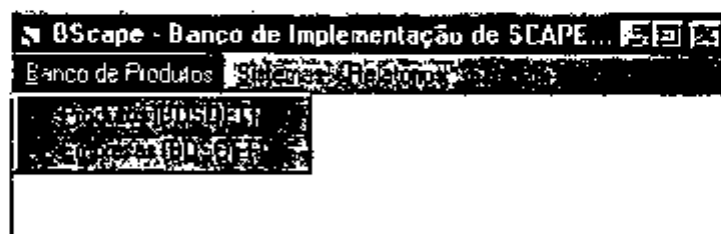


Figura D-1 - Cadastrando produtos e empresas no Banco de Produtos

#### b) Importando o diagrama de classes

O diagrama de classes produzido no programa CASE WITH-CLASS pode ser transferido automaticamente para o programa BSCAPE. Para isto deve-se gerar um arquivo de transferência no programa WITH CLASS definido pelo roteiro WC2BSCAP.SCT. Para realizar esta operação consulte o manual do programa CASE (MICROGOLD, 1996).

O arquivo gerado é lido pelo programa BSCAPE ao se selecionar a opção Sistemas >> importar no menu principal, como mostra a figura abaixo. O programa pede o nome do arquivo de transferência e cria um novo sistema para ser implementado no programa BSCAPE.

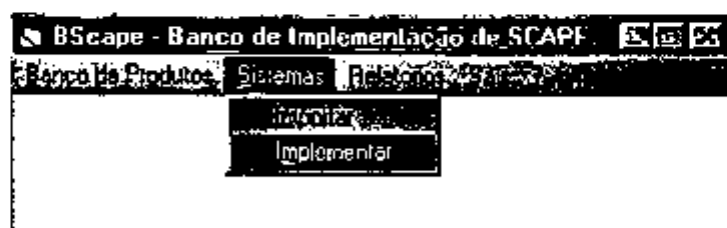


Figura D-2 - Opções do menu Sistemas no Programa BSCAPE.

### c) Selecionando o sistema a ser implementado

Antes de iniciar a implementação deve-se indicar para o programa qual será o sistema a ser implementado. Após acionar a opção Sistemas>>Implementar no menu principal, o programa apresenta uma interface que permite a navegação pelos sistemas armazenados e a seleção de um deles. A figura abaixo exemplifica esta interface.

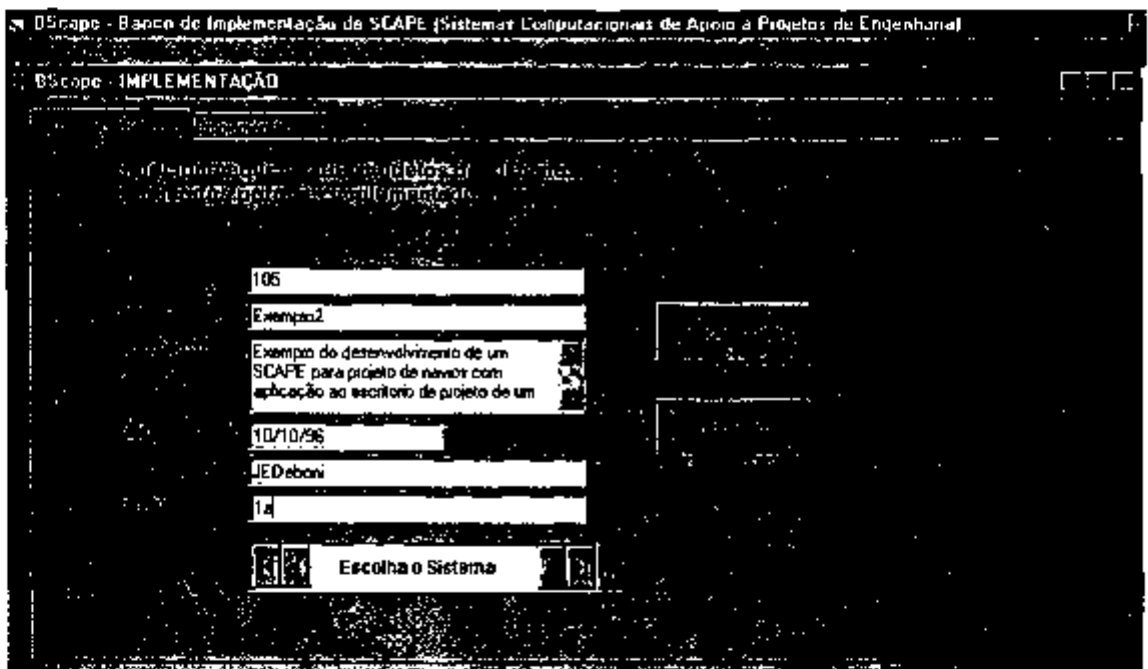


Figura D-3 - Interface para seleção do sistema

### d) Implementando uma função do diagrama

Selecionando-se a pasta "Implementação" o programa exibe a janela abaixo onde temos:

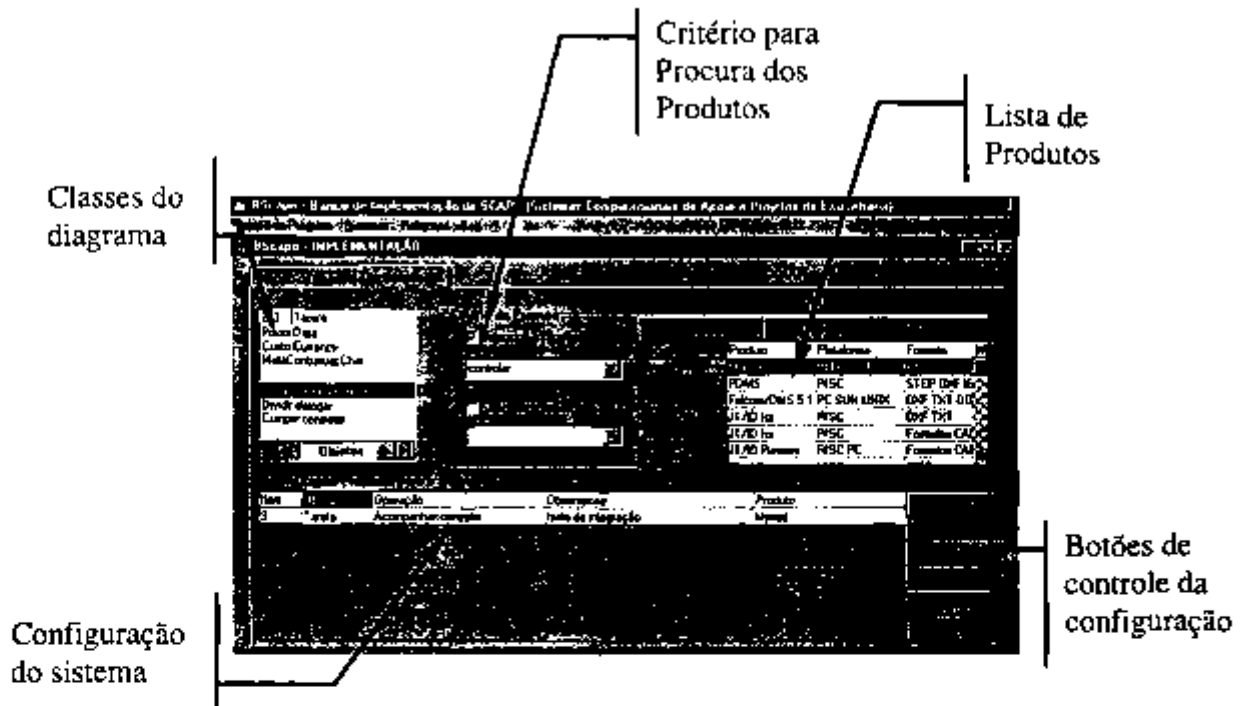


Figura D-4 - Interface para seleção do produto

Inicialmente se seleciona a classe a ser implementada através das setas imediatamente abaixo das classes do sistema (1). Com um Clique seleciona-se uma função da classe(2). Define-se um critério de procura escolhendo uma função e uma área de aplicação (3). Procura-se no Banco de Produtos os produtos que atendem a este critério (4), que são exibidos na lista de produtos. Seleciona-se um produto da lista com um Clique sobre seu número (5). Inclui-se o produto na configuração pressionando o botão incluir(6). Repete-se este processo para todas as funções e classes do diagrama.

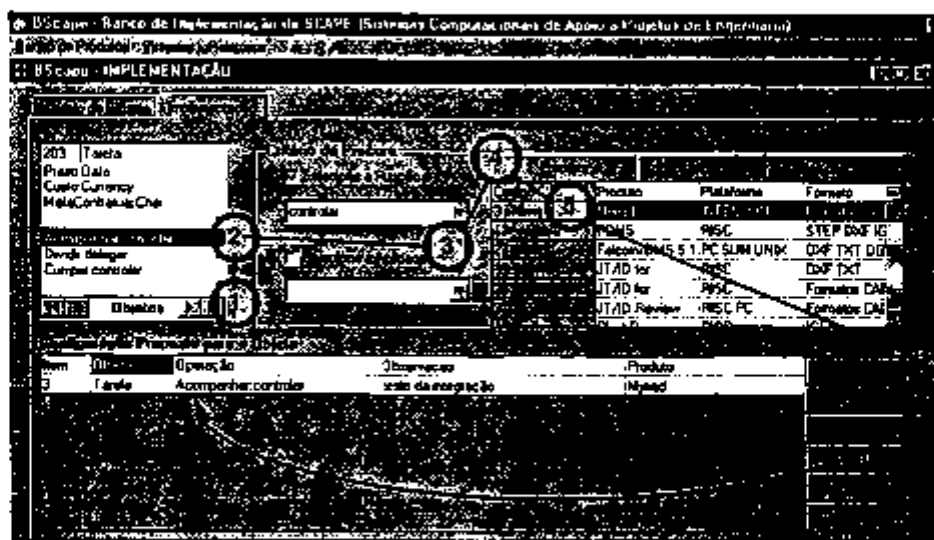


Figura D-5 - Sequência de operações para implementar uma função

#### e) Emitindo os relatórios da implementação

A opção relatórios do menu principal permite emitir diversos tipos de relatórios, que podem alternativamente, serem transferidos para editores de texto ou planilhas de cálculo.

#### 4. Conclusão

O programa BSCAPE constitui uma primeira tentativa de automação do processo de seleção de produtos para implementação de SCAPE. Diversos outros critérios podem ser criados, para aumentar a flexibilidade do processo. Uma tarefa importante é a atualização permanente do banco de produtos, que garante a disponibilidade do maior número possível de alternativas na escolha de produtos para reutilização.

### Referências Bibliográficas

- ADLER, P.S. CAD/CAM: Managerial Challenges and Research Issues. **IEEE Transactions on Engineering Management**. v.36, n.3, p. 202-215, aug. 1989.
- ANUÁRIO de Informática 96/97. IDG Computerworld do Brasil, Rio de Janeiro, 1996.
- ARAKAKI, R. **Uma Metodologia de Desenvolvimento e Implementação de sistemas CAE/CAD/CAM**. São Paulo, 1991. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo.
- BAZZO, W.A; PEREIRA L.T.V. **Introdução à Engenharia**. Florianópolis, Editora da UFSC, 3.ed. 1993, p 71-74.
- BERSON, A. **Client/Server Architecture**. McGraw-Hill Series on Computer Communications. 1992.
- BOOCH, G. Coming of age in an object-oriented world. **IEEE Software**, p.33-41, nov. 1994.
- BOOCH, G. **Object Solutions: Managing the Object-Oriented Project**. Menlo Park, 1996.
- BRUEN, J.D. Internationalization of CALS. **CALS Journal**. V.1 pg.21-25. Relatório n. 1497 1992. /Resumo do CERC, Concurrent Engineering Research Center extraído da INTERNET /
- BUONO, R. Pioneirismo no uso do CAE. **CADesign**, n. 2, Ano 1, 1995.
- CADRE Ed. **Introduction to Object-Oriented Modeling with OMT**. 1993. /White-paper, disponível na internet <http://www.cadre.com/>
- CADRE Ed. **ObjectTeam for OMT USER'S GUIDE**. Release 4.0, 1995. /Manual de programa de computador/
- CARTER, D.E; BAKER, B.S. **Concurrent Engineezing : The Product Development Environment for the 1990s**. Addison-Wesley. 1991.
- CERC. Repositório Eletrônico de Informações do Concurrent Engineering Research Center. Atualizado até 1993. /Endereço eletrônico: <http://www.cerc.wvu.edu/>

- CIM-OSA. Repositório Eletrônico de Informações do CIM - Open Systems Architecture. Atualizado até 1996.  
/Endereço eletrônico: <http://cimosa.cnt.pl/> /
- CLEMENTE, A.A. et al. Desenvolvimento de uma Metodologia de CAD/CAM para pequenas e médias indústrias. **Automação e Controle**, Janeiro, 1992.
- COAD, P. YOURDON, E. **Análise Baseada em Objetos**. Ed. Campus. Rio de Janeiro. 1993 (1a Ed em 1991)
- COAD, P. YOURDON, E. **Projeto Baseado em Objetos**. Ed. Campus. Rio de Janeiro. 1993 (1a Ed em 1991)
- CONGRESSO INTERNACIONAL DE COMPUTAÇÃO GRÁFICA, CINCONGRAF-94, São Paulo, 1994. **Anais**, SOBRACON ed. 1994.
- CONGRESSO INTERNACIONAL DE COMPUTAÇÃO GRÁFICA, CINCONGRAF-95, São Paulo, 1995. **Anais**, SOBRACON ed. 1995.
- CONGRESSO INTERNACIONAL DE COMPUTAÇÃO GRÁFICA; MÓDULO DE CAE/CAD/CAM, São Paulo, SOBRACON, 1995. **Anais**.
- CONGRESSO INTERNACIONAL DO COMPUTAÇÃO GRÁFICA, CICOMGRAF-94, São Paulo, SOBRACON, 1994. **Anais**.
- CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL, CONAI'94, São Paulo, 1994. **Anais**.
- CONGRESSO NACIONAL DE AUTOMAÇÃO, VI, CONAI 94, São Paulo, 1994. **Anais**. SUCESU. 1994.
- CONSTANTINE, L. Software Objectives. **Software Development**. v.3, n.6, p 87-88, June, 1995.
- COPESP. **Consolidação dos conceitos sobre SISTEC**. Relatório Interno. 1994.
- COPESP. **Implementação de um simulador para o SISTEC**. Relatório Interno. 1994.
- CUTKOSKY, M. et al. PACT: An Experiment in Integrating Concurrent Engineering Systems. **IEEE Computer**, p. 28-37, Jan. 1993.
- D'ISSY, M. "Upgrade" no Carro. **CADesign**, n. 5, Ano 1, p. 18-22, 1995.
- DE WYZE, J. Creating your CAD/CAM Strategy. **Computer Aided Desing Report**. v. 11, n 1-6, 1991. /Serie especial de artigos/

- DEJESUS, E. X. Programação sem sustos. **Byte**. Agosto, 1995.
- DIGITAL. **Open Systems Handbook: A Guide to Building Open Systems**. Digital Equipment Coporation. 1991.
- DUNCAN, D.A.R. **Um modelo para representação de atividades em aplicações de escritórios**. Porto Alegre, 1995. Tese (Doutorado). Universidade Federal do Rio Grande do Sul.
- EMBLEY, D; JACKSON, R.B.; WOODFIELD, S.N. OO systems analysis: is it or isn't it?. **IEEE Software**, p.19-33, jul. 1995.
- FALCONI, V. **TQC: Controle da qualidade total**. Rio de Janeiro, Bloch, 1992.
- GLASS; R.L. VESSEY; I. Contemporary application-domain taxonomies. **IEEE Software**, p.63-76, jul. 1995.
- GRAHAM, A. The CAD Framework Initiative. **IEEE Design and Test of Computers**. Sep. 1991.
- HAMMER, M.; CHAMPY, J. **Reengenharia, revolucionando a empresa**. Editora Campus. 12a edição. 1993.
- HIRAMA, K. **MetodOO- método de desenvolvimento de Sistemas Orientado a Objetos: Uma Abordagem Integrada à Análise Estruturada e Redes de Petri**. Sao Paulo, 1995. Tese (Doutorado) - Escola Politécnica, Universidade de Sao Paulo.
- HUMPHREY, W.S. Characterizing the Software Process: A Maturity Framework. **IEEE Software**. v.5, n.2., mar. 1988.
- IANNI, O.; **Mundialização Transfigura pessoas e idéias. O Estado de São Paulo**, São Paulo, 30 out. 1994.
- INTERGRAPH. Banco de referência sobre produtos de software, internet : <http://www.intergraph.com>. 1996.
- INTERGRAPH. Intergraph expands capabilities for ship design and production CAD/CAM systems. **Maritime Reporter**. Apr. 1994.
- JAIKUMAR, R. 200 years of CIM. **IEEE Spectrum**. Sept. 1993.
- KAPLAN, G. ed. Modern Design. **IEEE Spectrum**, Sept. 1993. Special Issue.
- KARINTHI, R. et al. **Promoting CE through Information Sharing**. Concurrent Engineering Research Center (CERC). Relatório n. 1490. /Resumo/

- KUSIAK, A.; WANG, J. Decomposition of tthe Design Process. **Journal of Mechanical Design**, v.115, p.687-695, dec. 1993.
- LIKER, J.K.; FLEISCHER, M. Implementing Computer-Aided Design: The Transition of Nonusers. **IEEE Transactions on Engineering Managment**, v.36, n.3, p.180-190, aug. 1989.
- MACKEY,W.; CARTER, J.C. Measure the Steps to Success. **IEEE Spectrum**, jun. 1994.
- MAJCHRZAK, A.; SALZMAN, H. Introduction to the Special Issue: Social and Organizational Dimensions of Computer-Aided Design. **IEEE Transactions on Engineering Managment**, v.36, n.3, p.174-179, aug. 1989.
- MARTIN, J. **Princípios de análise e projeto baseados em objetos**. Rio de Janeiro, Campus, 1994.
- MARTINI, J.S.C.; DEBONI, J.E.Z. Gerenciamento de Documentos em Ambiente de Engenharia Concorrente. In: CONGRESSO INTERNACIONAL DE COMPUTAÇÃO GRÁFICA, CINCONGRAF-94, São Paulo, 1994. **Anais**, SOBRACON ed. 1994a.
- MARTINI, J.S.C.; DEBONI, J.E.Z. Transferência Eletrônica de Informações Técnicas de Projetos. In: CONGRESSO NACIONAL DE AUTOMAÇÃO, VI, CONAI 94, São Paulo, 1994. **Anais**. SUCESU. 1994b.
- MARTUCCI JR, M. **Estudos de Estruturas de Sistemas de Automação**. São Paulo, 1992. Tese (Livre Docência). Escola Politécnica, Universidade de São Paulo.
- MAYER, R. J. et al. **Information Integration for CE (IICE) IDEF3 Process Description Capture Method Report**. Concurrent Engineering Research Center. Relatório n. 1494. May 1992. /Resumo/
- MAYER, R. J. et al. **Information Integration for CE (IICE) IDEF4 Object-Oriented Method Report**. Concurrent Engineering Research Center. Relatório n. 1495. May 1992. /Resumo/
- MICROGOLD ed. **WITH CLASS User's Guide**. 1996. /Manual de programa/
- MICROSOFT ed. **ACCESS 2.0 Guia do Usuário**. 1994. /Manual do programa/
- MICROSOFT ed. **Visual Basic 4.0 User's guide**. 1995 /Manual do programa/

- MILLS, R. E BECKERT, B. A. Software Tools. **Computer-Aided Engineering Special Report on Concurrent Engineering**. Out., 1991.
- MORGENSTERN, M. A Database Integration Platform for Concurrent Engineering. In: First Workshop on Enabling Technologies for Concurrent Engineering. Apr. 1992. **Resumos**. Concurrent Engineering Research Center. Relatório n. 1551.
- ORFALI, R.; HARLEY, D. Client/Server with Distributed Objects. **Byte**. v.20, n.4, p.151-162, April, 1995.
- PENNINO, T.P.; POTECHIN, J. Design for Manufacture. **IEEE Spectrum**. Sept. 1993.
- PLONSKI, G. A. The Need to Reconceptualize CAD: The Case of Brazilian Engineering Consultancy Firms. **IEEE Transactions on Engineering Management**, v.36, n.4. p.293-297, Nov. 1989.
- POUNTAIN, D. E SZYPERSKI, C. Extensible Software Systems. **Byte**. v.19, n.5, may, 1994.
- PRESSMAN, R.S. **Engenharia de software**. São Paulo, 3a ed, Makron books, 1995.
- PRIETO\_DIAZ, R. FREEMAN, P. **IEEE Software**. p.6-16, jan. 1987.
- PUJATTI, L. **Reutilização de Software: Aspectos da Tecnologia de Composição e ambiente de reutilização**. Sao Paulo, 1994. Dissertação (Mestrado) - Escola Politécnica, Universidade de Sao Paulo.
- REDDY, R.; WOOD, R.T.; CLEETUS, K.J. The DARPA initiative : Encouraging new practices. **IEEE Spectrum**, Jul. 1991.
- REDDY, Y.V.R. et al Computer Support for Concurrent Engineering. **IEEE Computer**, v. 26. p.12-16, Jan. 1993. Special Issue.
- ROSENBLAT, A. Ed. Concurrent Engineering. **IEEE Spectrum**, Jul. 1991. Special report.
- ROZENFELD, H; TAKAHASHI, S. RENTES, A.F. Modelamento de Empresas: Requisito para a Manufatura Integrada por Computador (CIM). In: CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL, 5. São Paulo, 1992. **Anais**. 1992.
- RUMBAUGH, et al. **Modelagem e Projetos Baseados em Objetos**. Rio de Janeiro, Campus, 1994.

- SCALA, R.M.; VAQUERO, H. Information integration in computer integrated manufacturing. **Revista Brasileira de Ciências Mecânicas**. v.17, n.1, p.35-49, 1995.
- SCHWAB et al. Concurrent Engineering. **IEEE Spectrum**, Sept. 1993.
- SHINA, S.G. New Rules for World-Class Companies. **IEEE Spectrum**, Jul. 1991.
- SHLAER, S; MELLOR, S. **The Shlaer-Mellor Method**. Project Technology, Berkley, CA, 1993.
- SILICON GRAPHICS. **Applications Directory**. 94/95 Edition. EUA. 1994. /Catálogo de produtos/
- SOBRACON Ed. **Guia Brasileiro de Automação Industrial**. 1992.
- SPRAGUE, R.A.; SINGH, K.J. E WOOD, R.T.; Concurrent Engineering in Product Development. **IEEE Design & Test of Computers**. p.6-13, Março, 1991.
- TANOMARU, N. **Projeto de Concepção e Projeto Básico do nível de coordenação de um sistema aberto de automação, utilizando conceitos de orientação a objetos**. São Paulo, 1994. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo.
- TEICHOLZ, E. ed. **CAD/CAM Handbook**. New York, McGraw-Hill, 1985.
- UDELL, J. COMPONENTWARE. **BYTE**. p.46-56, May, 1994.
- WEISBERG, D. E. Who's watching tomorrow's technology? **Computer Solutions**, 1995. Special issue /publicado para o AEC SYSTEMS'95/