

JORGE LUIS RISCO BECERRA

**APLICABILIDADE DO PADRÃO DE
PROCESSAMENTO DISTRIBUÍDO E ABERTO
NOS PROJETOS DE SISTEMAS ABERTOS DE
AUTOMAÇÃO**

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

São Paulo

1998

JORGE LUIS RISCO BECERRA

**APLICABILIDADE DO PADRÃO DE
PROCESSAMENTO DISTRIBUÍDO E ABERTO
NOS PROJETOS DE SISTEMAS ABERTOS DE
AUTOMAÇÃO**

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

Área de Concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Moacyr Martucci Jr.

1998

**Com amor, para minha esposa
Marcia, que me acompanha
nestes caminhos da vida.**

Agradecimentos

Ao Prof. Dr. Moacyr Martucci Jr. pela grande amizade, orientação e permanente incentivo, que se tornaram importantes no desenvolvimento deste trabalho, e na minha vida profissional.

A Profa. Dra. Selma Melnikoff pelo constante apoio e amizade, e pelas oportunidades oferecidas que aprimoraram minha experiência profissional.

Ao Prof. Dr. José Sidnei Colombo pelo permanente estímulo e pelos conselhos dados nos momentos certos.

Ao pessoal do Laboratório de Sistemas Abertos (LSA), meu muito obrigado.

Aos colegas do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo pelo apoio e incentivo.

A meus familiares pela compreensão e estímulo.

A todas as pessoas que de alguma forma ajudaram para alcançar este grande objetivo de minha vida.

Sumário

Lista de Figuras

Resumo

Abstract

Capítulo 1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Objetivos	4
1.3 Abrangência	6
1.4 Motivação	7
1.5 Justificativa	9
1.6 Metodologia	10
1.7 Histórico da Pesquisa	11
1.8 Estrutura da Tese	13
Capítulo 2 Processamento Distribuído e Aberto	14
2.1 Padronização ODP	14
2.1.1 Justificativa da Padronização	14
2.1.2 Objetivo do Padrão ODP	15
2.1.3 Situação Atual da Padronização	17
2.1.4 Normas ODP	18
2.2 Fundamentos ODP	20
2.2.1 Conceitos Primários	20
2.2.2 Conceitos para Modelagem	22
2.2.3 Conceitos para Especificação	23
2.2.4 Conceitos para Arquitetura	24
2.2.5 Conceitos Referentes à Conformidade	25
2.3 Os Pontos de Vista	26
2.3.1 Definição	27
2.3.2 Formas de Utilização	28
2.3.3 Ponto de Vista da Empresa	31

2.3.4 Ponto de Vista da Informação.....	32
2.3.5 Ponto de Vista da Computação	33
2.3.5.1 Objeto Computação	38
2.3.5.2 Interfaces Computacionais.....	39
2.3.5.3 Tipos de Conexão	40
2.3.6 Ponto de Vista da Engenharia	42
2.3.6.1 Objeto Básico de Engenharia.....	42
2.3.6.2 Estrutura de Gerenciamento	42
2.3.6.3 Estrutura de Comunicação.....	46
2.3.7 Ponto de Vista da Tecnologia	47
2.4 Transparências e Funções.....	50
2.4.1 Transparências Distribuídas	50
2.4.2 Funções ODP	51
2.4.2.1 Funções de Gerenciamento.....	52
2.4.2.2 Funções de Coordenação	53
2.4.2.3 Funções dos Repositórios.....	54
2.4.2.4 Funções de Segurança	54
Capítulo 3 Metodologia de Projeto ODP	56
3.1 Modelo de Referência SAA	56
3.1.1 Características.....	58
3.1.2 Estrutura do SAA	59
3.1.2.1 Classes de Objetos Automação	59
3.1.2.2 Níveis Hierárquicos	60
3.1.2.3 Camadas Funcionais	62
3.2 Estrutura Metodológica ODP.....	65
3.2.1 Definição e Características	65
3.2.2 Fases de Desenvolvimento.....	68
3.2.3 Formas de Aplicação.....	71
3.3 Especificação ODP	72
3.3.1 Especificação do Ponto de Vista Empresa	72
3.3.1.1 Definição das Regras de Empresa.....	73
3.3.1.2 Modelagem do Ponto de Vista da Empresa	74

3.3.2	Especificação do Ponto de Vista Informação.....	75
3.3.2.1	Definição das Regras de Informação	75
3.3.2.2	Modelagem do Ponto de Vista da Informação	76
3.3.3	Especificação do Ponto de Vista Computação	77
3.3.3.1	Definição das Regras da Computação.....	77
3.3.3.1.1	Regras de Interação	78
3.3.3.1.2	Regras de Conexão	78
3.3.3.1.3	Regras de Estruturação	79
3.3.3.1.4	Regras de Falhas.....	80
3.3.3.2	Modelagem do Ponto de Vista Computacional	80
3.3.4	Especificação do Ponto de Vista da Engenharia	81
3.3.4.1	Regras de Engenharia	81
3.3.4.1.1	Regras dos Canais	82
3.3.4.1.2	Regras das Interfaces Referenciais	83
3.3.4.1.3	Regras de Conexão Distribuída	83
3.3.4.1.4	Regras de Relocalização	83
3.3.4.1.5	Regras da Estrutura de Gerenciamento.....	84
3.3.4.1.6	Regras de Falhas.....	84
3.3.4.2	Modelagem do Ponto de Vista da Engenharia.....	85
3.3.5	Especificação do Ponto de Vista Tecnológico	85
3.4	Arquitetura ODP	86
3.4.1	Considerações Iniciais.....	87
3.4.2	Estrutura da Arquitetura.....	89
3.4.2.1	Modelo Conceitual	89
3.4.2.2	Arquitetura Distribuída ODP	90
3.4.3	Funcionalidades ODP	92
3.4.3.1	Definição das Funções ODP	93
3.4.3.2	Regras das Transparências	94
3.5	Projeto ODP	95
3.5.1	Análise dos níveis hierárquicos	96
3.5.2	Projeto detalhado.....	97
3.6	Implementação.....	98

Capítulo 4 Desenvolvimento de um Sistema Aberto de Automação de Projetos	99
4.1 Considerações Gerais sobre o Experimento	99
4.2 Engenharia Simultânea e os Projetos dos Grandes Empreendimentos	103
4.2.1 Engenharia Simultânea	103
4.2.2 Ciclo de Vida Paralelo e a Equipe Virtual	106
4.2.3 Base de Dados Única de Projeto	109
4.2.4 Serviços do Ambiente de Projeto	114
4.2.5 Organização	117
4.2.6 Requisitos do Sistema	119
4.3 Modelo Hierárquico do Ambiente de Projeto	120
4.4 Modelos dos Cinco Ponto de Vista	125
4.4.1 Modelo da Empresa	125
4.4.2 Modelo da Informação	128
4.4.3 Modelo da Computação	131
4.4.4 Modelo da Engenharia	134
4.4.5 Modelo da Tecnologia	135
4.5 Arquitetura dos Níveis Hierárquicos	141
4.6 Especificação da Implementação	146
4.7 Análise dos Resultados	150
Capítulo 5 Considerações Finais	156
5.1 Conclusões	156
5.2 Comentários Gerais	158
5.3 Continuidade da Pesquisa	159
5.4 Contribuição Acadêmica	160

Lista de Figuras e Tabelas

Figura 2.1	Categorias de Conceitos ODP	21
Figura 2.2	Estrutura do Ponto de Vista da Empresa.....	34
Figura 2.3	Estrutura do Ponto de Vista da Informação	35
Figura 2.4	Estrutura do Ponto de Vista da Computação.....	36
Figura 2.5	Relação Entre os Objetos Computação e a Estrutura de Comunicação.....	37
Figura 2.6	Tipos de Conexão Computacional.....	41
Figura 2.7	Estrutura do Ponto de Vista da Engenharia.....	44
Figura 2.8	Estrutura Gerencial do Ponto de Vista da Engenharia.....	45
Figura 2.9	Estrutura da Comunicação do Ponto de Vista da Engenharia.....	48
Figura 2.10	Estrutura do Ponto de Vista da Tecnologia	49
Figura 3.1	Modelo de Referência SAA.....	57
Figura 3.2 a	Estrutura Hierárquica dos Objetos Automação.....	63
Figura 3.2 b	Instâncias dos Objetos Instrumentação e Coordenação	63
Figura 3.3	Camadas Funcionais do Nível Hierárquico de Controle	64
Figura 3.4	Estrutura da Metodologia ODP	69
Figura 3.5	Relação entre a Arquitetura ODP, a Especificação ODP, e o Modelo de Referência SAA	88
Figura 3.6	Modelo Conceitual da Arquitetura Distribuída	91
Figura 3.7	Arquitetura ODP	91
Figura 4.1	Elementos do Experimento.....	102
Figura 4.2 a	Ciclo de Vida Seqüencial.....	108
Figura 4.2 b	Ciclo de Vida da Engenharia Simultânea	108
Figura 4.3	Modelo Conceitual de Base de Dados única de Projeto.....	111
Figura 4.4	Modelo Físico da Base de Dados Única do Ambiente de Projeto..	113
Figura 4.5	Modelo dos Serviços do Ambiente de Projeto	115
Figura 4.6	Organização de um Ambiente de Projeto	118
Figura 4.7	Modelo Hierárquico.....	122
Figura 4.8	Diagrama de Objetos do Ponto de Vista da Empresa	127

Figura 4.9 Uses Cases de uma Regra da Empresa	127
Figura 4.10 Diagrama da Informação do Esquema Invariante.....	129
Figura 4.11 Diagrama de Colaboração do Esquema Estático	129
Figura 4.12 Diagrama de Estados de um Esquema Dinâmico	130
Figura 4.13 Contrato de uma Regra de Informação	132
Figura 4.14 Diagrama de Objetos do Ponto de Vista Computacional.....	133
Figura 4.15 Uses Cases de uma Regra da Computação.....	133
Figura 4.16 Contrato de uma Regra de Computação	136
Figura 4.17 Diagrama de Objetos do Ponto de Vista da Engenharia	137
Figura 4.18 Uses Case de uma Regra da Engenharia	138
Figura 4.19 Contrato de uma Regra de Engenharia.....	139
Figura 4.20 Diagrama de Objetos do Ponto de Vista da Tecnologia	140
Figura 4.21 Estrutura dos Níveis Hierárquicos do Ambiente de Projeto	144
Figura 4.22 Arquitetura do SAA de Projeto.....	145
Figura 4.23 Estrutura do Sistema Aberto de Automação de Projetos.....	147
Figura 5.1 Estrutura de Pesquisa	161
Tabela 2.1 Exemplos de Forma de Utilização dos Pontos de Vista.....	30
Tabela 3.1 Resumo da Estrutura da Metodologia ODP	70
Tabela 4.1 Resultados do Experimento ODP	154

Resumo

Esta tese visa demonstrar a aplicação do padrão ODP (Open Distributed System) no desenvolvimento de sistemas distribuídos e abertos aplicados em automação.

Para atingir este objetivo define-se uma estratégia cujos elementos básicos são: a metodologia ODP e o experimento ODP. A metodologia suporta o padrão ODP, numa estrutura sistemática e o experimento, aplica-o num projeto, para observar o comportamento do processo de aplicabilidade.

A metodologia ODP está composta dos conceitos, procedimentos e modelos, propostos pelo padrão ODP; das técnicas e conceitos da orientação a objetos e da estrutura de um modelo de sistema aberto de automação (SAA), constituindo uma estrutura coerente e gerenciável, de aplicação prática nos projetos de sistemas abertos em automação.

No experimento ODP, a metodologia é aplicada no desenvolvimento de um sistema de automação, cujo escopo são os projetos dos grandes empreendimentos, que utilizam em seu processo de desenvolvimento a engenharia simultânea. O resultado desta automação será caracterizada pela obtenção de um sistema distribuído, aberto e configurado como ambiente cooperativo (Groupware) de projeto.

Finalmente, os resultados do experimento serão comparados com outros tipos de metodologias que não utilizam o ODP, para poder concluir sobre a utilização do padrão ODP nos projetos de sistemas de automação.

Abstract

This thesis purposes to demonstrate the applicability of the open distributed processing standard, in the development of the open distributed system applied in the automation.

To achieve this objective a strategy is defined, whose basic elements are the ODP methodology and the ODP experiment. The methodology supports the ODP standard in a systematic structure and the experiment applies it in one project, to observe the behavior of the applicability process.

The ODP methodology is composed by concepts, procedures and models, proposed by the ODP standard; by techniques and concepts of the objects-oriented, and the structure of an open automation system model (OAS), establishing an manageable and coherent structure.

In the ODP experiment, this methodology is applied in the development of an automation system which scope is the large enterprise projects that use in his development process the concurrent engineering. The result of this automation will be characterized by the Open Distributed System and will be configured as a project cooperative environment (Groupware).

Finally, the experiment results will be compared with other methodologies that don't use ODP, to obtain conclusions about the use of the ODP standard in the automation system projects.

Capítulo 1 Introdução

Este capítulo apresenta os subsídios necessários, que definem o contexto geral, da tese intitulada de “Aplicabilidade do Padrão de Processamento Distribuído e Aberto nos Projetos de Sistemas Abertos de Automação”.

O contexto geral deste capítulo, está composto de tópicos referentes aos objetivos, abrangência do trabalho, motivação, justificativa, histórico da pesquisa e a metodologia utilizada

1.1 Considerações Iniciais

Os grandes empreendimentos são caracterizados por serem constituídos de múltiplos projetos dependentes, processos, sistemas e equipes [Martucci, 92].

Os grandes empreendimentos, por estarem constituídos de muitos projetos, processos, sistemas, caracterizam-se principalmente por possuir grandes tempos de desenvolvimentos, normalmente entre 1 e 12 anos, onde existe a participação de dezenas e até várias centenas de pessoas, com custos que podem atingir até centenas de milhões de dólares [Ryan, 92].

Este tipo de projeto exige a implantação de uma estrutura que possibilite a realização das tarefas de controle, supervisão, coordenação e gerenciamento de forma eficiente, para poder atingir benefícios em três aspectos importantes: qualidade, custo e tempo de desenvolvimento, [Stevens, 87], [Ryan, 92], [Bailetti, 94], [Bernstain, 96].

No desenvolvimento deste tipo de projetos ocorrem uma série de problemas típicos como por exemplo: inconsistências nos elementos dos sistemas,

interferências indesejáveis nos arranjos físicos, erros e reprojets no desenvolvimento.

Estes problemas e outros relacionados com áreas envolvidas no projeto, produzem diretamente aumento nos tempos de desenvolvimento, aumento nos custos do projeto e finalmente diminuem a qualidade do produto ou do serviço final, prejudicando o projeto [Martucci, 92], [Handfield, 94].

Desta forma, e para este caso de grandes empreendimentos, existe a necessidade de uma estrutura de desenvolvimento de projeto adequada, onde devem estar três elementos importantes: o sistema de automação, a estrutura de suporte da informação e a metodologia do processo de desenvolvimento.

O sistema de automação é uma estrutura composta de um conjunto de equipamentos de automação e elementos da tecnologia de informação, onde deve existir um fluxo contínuo das informações de forma a obter ganhos na produtividade e na homogeneização da qualidade, [Becerra, 96a]

A estrutura de suporte da informação está composta de elementos que permitem a manipulação, gerenciamento e armazenamento da informação de forma adequada, disponibilizando a informação de forma contínua em todos os domínios da empresa.

Com o objetivo de criar um sistema coerente e produtivo, adiciona-se aos elementos anteriores, a metodologia do processo de desenvolvimento que permitirá o gerenciamento e o controle do projeto como um todo.

No contexto desta tese, a metodologia de processo de desenvolvimento utilizada é a engenharia simultânea, que permitirá a execução das fases do projeto de forma paralela e síncrona, visando a diminuição dos tempos de

desenvolvimento e a redução dos erros de projeto e como consequência obter custos competitivos no mercado e qualidade adequada, [Gatenby, 94].

O ambiente de suporte da engenharia simultânea está caracterizado pela execução de atividades cooperativas, isto é, existe uma ambiente cooperativo computacional (Groupware) e também possui um eficiente sistema aberto de informações [Becerra, 96b], [Rodden, 92].

A aplicação da engenharia simultânea coloca novos requisitos na especificação do sistema de automação aumentando sua complexidade, que por sua vez, exige uma metodologia de projeto adequada que permita a correta definição da especificação do sistema. Atualmente, os métodos tradicionais de desenvolvimento de sistemas possuem dificuldades para definir os requisitos do sistema [Davis, 93], [Steven 87], [Navarro, 93].

Com estas considerações iniciais, será desenvolvido um sistema de automação de projeto especificado como um sistema distribuído e aberto com objetivo de obter altos graus de interoperabilidade e de cooperação entre os elementos que o compõem.

O sistema aberto de automação a desenvolver seguirá o modelo proposto em [Martucci, 92], denominado de Sistema Aberto de Automação (SAA), e utilizará uma metodologia baseada no modelo de referência da ISO (International Organization for Standardization) denominado de ODP (Open Distributed Processing), [ISO, 95a], [ISO, 95b], [ISO, 95c], [ISO, 95d], [ISO, 95e].

O trabalho visa também a obtenção de conclusões importantes sobre a aplicação deste padrão, especialmente na área de automação, que poderão ser estendidos às outras áreas dos sistemas distribuídos e abertos, e desta forma gerar contribuições na comunidade científica e técnica.

1.2 Objetivos

O primeiro objetivo e o mais importante deste trabalho de tese é demonstrar a aplicação do padrão ODP no desenvolvimento de sistemas de automação distribuídos e abertos.

Este objetivo abrange a determinação da estratégia de aplicação do padrão, a determinação dos benefícios e a detecção dos problemas encontrados na utilização do padrão ODP.

Com relação à estratégia de aplicação do padrão ODP, deve-se determinar métodos e esquemas, como uma consequência da aplicação dos conceitos, procedimentos e modelos sugeridos pelo padrão ODP.

No que diz respeito aos benefícios obtidos na aplicação do padrão ODP, eles serão analisados nos seguintes aspectos: dentro do escopo da definição dos requisitos do sistema; como elemento de um processo metodológico; seu impacto nas atividades gerenciais; e, finalmente, a sua relação com a garantia da qualidade.

Com referência à detecção dos problemas na aplicação do padrão ODP, estes serão analisados e serão definidas soluções, que podem resultar na inclusão de métodos, procedimentos e outros conceitos que permitam a sua viabilização.

Finalmente, visa obter conclusões nos aspectos técnicos, gerenciais, e de qualidade.

O segundo objetivo, deste trabalho de tese, visa a proposta de uma metodologia de desenvolvimento de projeto de sistemas abertos de automação baseados no padrão ODP.

Esta atividade consiste em definir uma estrutura sistemática e seqüencial cujas fases do desenvolvimento suportam de forma adequada os conceitos, procedimentos e modelos definidos no padrão ODP, adicionando-se outros conceitos, procedimentos e modelos de forma a obter uma estrutura coerente e de fácil utilização.

Este objetivo origina-se por causa que o padrão ODP representa um modelo de referência abrangente, não pertence a um domínio específico, nem muito menos representa uma metodologia de desenvolvimento.

Como conseqüência do alcance deste objetivo serão apresentados comentários e contribuições que permitam a aplicação desta metodologia para outros domínios de aplicação diferentes da automação.

O terceiro objetivo do trabalho de tese abrange a definição de uma arquitetura aberta específica para ambientes que suportam a engenharia simultânea.

Esta estrutura específica, conseqüência do objetivo anterior, deverá incluir todos os requisitos de um sistema distribuído, aberto e cooperativo, desta forma, poderá ser utilizada em diferentes implementações de ambientes que suportam a engenharia simultânea.

Na definição desta arquitetura, utiliza-se a especificação ODP, os métodos definidos nas metodologias orientadas a objetos, e será uma opção de arquitetura de sistemas distribuídos, como são as propostas CORBA, DCE, entre outras, como pode-se ver em [Tanenbaum, 95], [Crowcroft, 95], [Schill, 90].

O quarto objetivo deste trabalho de tese representa o aspecto prático da Tese, será definida a especificação da implementação do sistema aberto de

automação, indicando os equipamentos e as tecnologias abertas e proprietárias a serem utilizados na implementação.

Esta especificação determinará a viabilidade de implementação do sistema aberto de automação de projetos. Os requisitos mínimos de implementação poderão ser estendidos para qualquer ambiente que suporte engenharia simultânea em suas diversas formas de aplicação.

Finalmente, o quinto objetivo está dentro do contexto acadêmico e de pesquisa, dever-se-á impulsionar as atividades de ensino, pesquisa, desenvolvimento e extensão do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo.

Este objetivo será atingido, determinando e apresentando novas frentes de pesquisa, e divulgando os conhecimentos estruturados neste trabalho, através das atividade de ensino e extensão.

1.3 Abrangência

O escopo deste trabalho de tese, estará restrito ao contexto definido pelo projeto de um sistema aberto de automação aplicado a projetos dos grandes empreendimentos, cujo processo de desenvolvimento utiliza a engenharia simultânea.

Com relação aos resultados obtido na aplicação do ODP, serão apresentados resultados do tipo qualitativo e heurístico, este trabalho de tese não visa a apresentação de métricas específicas.

A implementação do sistema limitar-se-á ao nível de especificação, pois sua complexidade estrutural e seu alto custo inviabiliza esta implementação no âmbito acadêmico.

A especificação da implementação deverá definir as tecnologias que implementam os elementos que compõem a arquitetura, ressaltando as soluções padronizadas tanto de fato como as de júri.

Em relação ao modelo de sistema aberto de automação a utilizar, limitar-se-á ao escopo do modelo de sistema aberto de automação (SAA), cuja estrutura hierarquizada e distribuída será a base do sistema ODP, que será definido neste trabalho.

As versões das normas do padrão ODP a utilizar, serão as indicadas nas seguintes referências bibliográficas [ISO, 95a], [ISO, 95b], [ISO, 95c], [ISO, 95d], [ISO, 95e], desta forma, pretende-se conseguir um desenvolvimento homogêneo evitando os impactos que possam originar uma nova versão do padrão.

É bom ressaltar, que o escopo desta tese não engloba a utilização da norma quatro do padrão ODP, denominado de semântica da arquitetura. Esta parte da norma apresenta definição dos conceitos ODP utilizando linguagens de descrição formal, [Najm, 95].

Com relação aos conceitos e aspectos da metodologia orientada a objeto utilizada neste trabalho, serão utilizados os apresentados nas referências [Rumbaugh, 91], [UML, 97], ressaltando que a referência [Rumbaugh, 91] foi recomendada pelo padrão ODP.

1.4 Motivação

A motivação deste trabalho de tese, foi encontrada em quatro assuntos que merecem destaque: a utilização do padrão ODP na área de automação, a pesquisa em torno dos sistemas abertos de automação, a evolução do

Laboratório de Sistema Abertos (LSA) da Escola Politécnica da Universidade de São Paulo como centro de pesquisa e ensino na automação, e, finalmente, a preocupação de orientar à comunidade industrial na utilização de novas tecnologias de informação.

O motivo principal está na utilização do padrão ODP na área da automação, motivação originada pela existência de um forte movimento referente ao padrão ODP dentro da comunidade científica, existem hoje trabalhos que estão sendo dirigidos ao desenvolvimento de sistemas de multimídia e de telecomunicações [Rahkila 97], [Valérie, 95], [Magedanz, 93].

Estas atividades criaram um grande interesse e expectativa sobre a importância do padrão, e existe a necessidade de comprovar sua capacidade em outros domínios de aplicação como por exemplo os sistemas de automação de projetos [Meer, 95].

Outra fonte de motivação, é a pesquisa em sistemas abertos iniciada no trabalho de [Martucci, 92], e que originou o primeiro trabalho de pesquisa do autor (nível de mestrado), onde estudou-se a aplicação de arquiteturas abertas no projeto de equipamentos de automação [Becerra, 94].

Dentro dessa linha de pesquisa, se está dando continuidade a esta evolução com a proposta deste trabalho de tese que visa a estruturação de uma metodologia de projeto baseado no padrão ODP aplicado a sistemas abertos de automação.

Outra fonte de motivação é a evolução do Laboratório de Sistemas Abertos do Departamento de Engenharia de Sistemas Digitais da Escola Politécnica da Universidade de São Paulo, onde o autor participa ativamente em diferentes atividades de pesquisa, ensino e extensão.

O desenvolvimento de sistemas abertos de automação são assuntos de interesse na comunidade, sua aplicação exige dos usuários profundos conhecimentos sobre diferentes tipos de tecnologia, e neste contexto, o autor motiva-se para propor uma solução prática que possa ser entendida e utilizada de forma adequada.

A comunidade, composta de empresas, são os verdadeiros usuários de tecnologia da informação, existe a necessidade de orientar e contribuir da divulgação de novas tecnologias objetivando o aumento do nível tecnológico do parque industrial contribuindo desta forma no papel que a universidade exerce dentro da comunidade.

1.5 Justificativa

A grande importância do padrão ODP, no desenvolvimento de sistemas distribuídos e abertos, cria a justificativa necessária para poder realizar esta tese, pois é necessário aprofundar-se no entendimento e aplicação deste padrão com o objetivo de atingir sua consagração.

Através da aplicação do padrão ODP na área de sistemas de automação, dever-se-á encontrar conclusões importantes que permitam fortalecer a aceitação, consagração e estabilização do padrão.

Os resultados desta tese devem gerar novas informações que serão utilizados no desenvolvimento de pesquisa em diferentes níveis acadêmicos dentro da linha denominada de sistemas distribuídos e abertos. Esta razão cria outra justificativa da tese.

Os modelos de arquiteturas distribuídas e abertas [Stefani, 95], [Rubin, 94] estão sendo utilizados de forma gradual no mercado atual, existe a

necessidade de incentivar, ainda mais, esta utilização através de propostas práticas.

Este contexto origina outra justificativa da tese, pois como consequência deste trabalho será apresentada uma proposta prática de desenvolvimento de um sistema ODP que poderá ser utilizado como referência básica em outras aplicações.

No mesmo contexto, a arquitetura aberta de automação de projetos cujo processo baseia-se na engenharia simultânea, produto desta tese, poder-se-á tornar um modelo referencial que poderá ser aplicado em outros domínios do mercado.

1.6 Metodologia

Para o desenvolvimento desta tese e com objetivo de gerenciar a evolução do processo visando as metas e a qualidade, foi utilizada uma metodologia cujas fases são definidas a seguir:

- Pesquisa: fase que engloba a seleção das informações as quais foram estruturadas em grupos, cada grupo corresponde a um assunto relevante da tese. Esta atividade é realizada ao longo do processo de desenvolvimento sendo que esta diminui perto do final do projeto;
- Estruturação da metodologia ODP: fase que abrange as atividades referentes à formação da proposta e a definição da metodologia de projeto baseado no padrão ODP, determinação das fases e procedimentos;
- Análise do sistema: fase que engloba as atividades de caracterização do sistema de automação de projeto, a modelagem do sistema em cinco pontos de vista e especificação do sistema;

- Definição da arquitetura: fase que engloba a proposta de uma arquitetura que represente o sistema aberto de automação de projetos; aqui deverão ser analisadas as várias propostas de arquitetura e, segundo a especificação da fase anterior, definir uma arquitetura específica;
- Implementação: fase que engloba a especificação da implementação, pois o sistema com um todo possui uma complexidade muito grande, que inviabiliza sua implementação física num laboratório acadêmico.

1.7 Histórico da Pesquisa

Este item tem por objetivo, apresentar a evolução da pesquisa sobre sistemas abertos aplicados a sistemas de automação realizada no Laboratório de Sistemas Abertos (LSA) do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo, e que envolve o trabalho de tese do autor.

O início da pesquisa encontra-se no trabalho de [Martucci, 92] que propõe a utilização do conceito de sistemas abertos na automação e apresenta o modelo de sistema aberto de automação (SAA), cuja estrutura hierárquica poderia ser aplicada em vários segmentos da indústria.

Os próximos trabalhos dentro desta linha de pesquisa foram focalizados para a demonstração dos conceitos e do modelo proposto. O trabalho de [Becerra 93] demonstra a utilização dos conceitos de sistemas abertos na definição de arquiteturas abertas aplicadas a equipamentos de automação, proposta que esta sendo adotada como uma solução adequada na indústria [Proctor, 97].

O trabalho de [Tanomaru, 94], preocupou-se pela experimentação do modelo SAA e pela definição de seu nível de coordenação, utilizando o paradigma de orientação a objetos na fase de análise e projeto.

O trabalho de [Matsuyama, 97] conclui o trabalho de experimentação do modelo SAA, obtendo resultados de aplicação nas áreas de automação bancária, processos contínuos e discretos e automação predial. Utilizou a mesma metodologia de projeto do trabalho anterior.

Com estes subsídios, originou-se a necessidade de propor uma metodologia de projeto adequada para o projeto de sistemas abertos e distribuídos que apresente coerência desde a especificação até a implementação do sistema.

Visando esta necessidade, inicia-se o trabalho de [Silva, 96], que apresenta uma proposta de sistema de informação cuja estrutura suporta como metodologia de processo a engenharia simultânea. Este trabalho determinou a complexidade do sistema de automação de projeto, escopo do experimento que faz parte do trabalho desta tese.

A seleção deste caso especial foi realizada porque a engenharia simultânea exige um sistema de automação aberto, distribuído e cooperativo o que termina por aumentar a complexidade dos requisitos do projeto de sistema aberto.

Continuando nessa linha de pesquisa, este trabalho de tese visa propor uma metodologia de projeto de automação, cujo ponto central está na utilização do padrão ODP no ciclo de vida do projeto, e utilizará os resultados da análise e projeto do trabalho de [Tanomaru, 94] e as conclusões no que se refere ao modelo SAA do trabalho de [Matsuyama, 97].

Estas pesquisas foram realizadas no Laboratório de Sistemas Abertos (LSA) cujo objetivo principal é o estudo e aplicação de padrões nos sistemas de informação.

1.8 Estrutura da Tese

Este trabalho de tese abrange basicamente a definição de uma metodologia de projeto ODP e o experimento desta metodologia na definição de um sistema aberto de automação que suporta engenharia simultânea. A estrutura da tese está composta de cinco capítulos, cujos conteúdos serão definidos a seguir:

- Primeiro capítulo: apresenta-se as razões que justificam o desenvolvimento desta tese assim como a definição dos objetivos que serão atingidos;
- Segundo capítulo: neste capítulo apresenta-se uma introdução ao padrão ODP, onde serão apresentados os conceitos e estruturas básicas;
- Terceiro capítulo: será apresentado a metodologia ODP, onde serão apresentados as fases, procedimentos e modelos;
- Quarto capítulo: apresenta a especificação do ambiente cooperativo que suporta engenharia simultânea, o experimento que valida a metodologia ODP, a especificação da implementação da arquitetura e os resultados;
- Quinto capítulo: apresenta as conclusões gerais da tese, os trabalhos que poderão ser realizados no futuro dentro do contexto desta tese, comentários finais e a contribuição acadêmica atingida.

A seguir, apresenta-se os fundamentos básicos referentes ao padrão ODP, importantes para o entendimento do desenvolvimento da tese.

Capítulo 2 Processamento Distribuído e Aberto

Neste capítulo será apresentado o embasamento teórico da Tese, e contém os conhecimentos básicos para o entendimento da aplicação do padrão ODP (Open Distributed Processing).

Especificamente, este capítulo contém um tratado teórico sobre o padrão ODP, sem a formalidade que caracteriza um padrão, mas estruturado de forma a permitir um fácil entendimento, e conseqüentemente facilitar outras pesquisas e aplicações sobre esta área.

A estrutura deste capítulo traz os seguintes tópicos: a padronização ODP, conceitos, arquitetura e as transparências.

2.1 Padronização ODP

O movimento de padronização ODP iniciado pela ISO (International Organization for Standardization) e a ITU (International Telecommunication Union) no ano 1988 procura a solução de diferentes problemas originados nos sistemas de informação [Farooqui, 95], [Bowen, 91].

2.1.1 Justificativa da Padronização

Os sistemas de informação são estruturas distribuídas cujos elementos são processadores de informação interagindo através de interfaces definidas. Estas estruturas caracterizam-se por estar composto de elementos heterogêneos, onde existem problemas de integração que dificultam o fluxo contínuo da informação [Nankman, 96], [Nicol, 93].

Os problemas de integração entre os elementos heterogêneos foram sendo solucionados com a aplicação de graus de interoperabilidade no sistema,

configurando sistemas distribuídos e cooperativos, onde os serviços (recursos) são acessados adequadamente e onde podem ser implantados novos serviços de forma simples, tornando os sistemas eficientes e altamente produtivos [Becerra, 97].

Dentro deste contexto, os sistemas abertos tem contribuído fortemente na integração dos sistemas de informação; baseados na implementação de elementos cujas especificações seguem padrões internacionais e onde a preocupação principal é a implantação de altos graus de interoperabilidade, transformou-se numa alternativa tecnológica em diferentes domínios de aplicação [Becerra, 96a], [Navarro, 93], [ITU, 96].

Desta forma, os sistemas distribuídos e abertos são motivos de pesquisas e propostas tecnológicas, merecendo grande destaque o desenvolvimento deste tipo de sistemas. Dentro desta orientação, existem duas preocupações importantes, a definição de arquiteturas distribuídas e abertas e a necessidade de ambientes para desenvolver sistemas distribuídos e abertos [Scherer, 97], [Rosa, 95], [Barr, 93].

A ISO e a ITU encontraram nos parágrafos anteriores, as justificativas e motivações suficientes para propor o modelo de referência ODP cuja estrutura visa suportar as especificações de sistemas distribuídos e abertos e visa o desenvolvimento de padrões para sistemas distribuídos.

2.1.2 Objetivo do Padrão ODP

O objetivo principal do padrão ODP é suportar a criação de sistemas distribuídos e abertos, onde exista interoperabilidade entre diversos sistemas, sejam sistemas ODP, ou sistemas proprietários ou outros sistemas padronizados do mercado.

Para atingir este objetivo, o padrão ODP apresenta uma estrutura para especificar padrões ODP e elementos e componentes do sistema, desta forma os sistemas informação construídos segundo estas estruturas e padrões ODP, tornam-se sistemas distribuídos e abertos.

Estes sistemas distribuídos e abertos, construídos segundo o contexto ODP, visam possuir as seguintes propriedades [Becerra, 97], [Tschanmer, 94], [ISO, 95a]:

- Aberto: a utilização de padrões de fato e de júri viabilizam a implantação desta propriedade e visam a obtenção de altos graus de portabilidade e de interoperabilidade dentro do sistema;
- Integração: esta propriedade permite a incorporação de vários sistemas e recursos de forma simples e a baixo custo, com objetivo de compor um sistema altamente produtivo e eficiente;
- Flexibilidade: com esta propriedade suporta-se a evolução do sistema, isto é, inclusão de novas tecnologias, configurações dinâmicas por causa de novas especificações de funcionamento, e configurações por causa da mobilidade física dos elementos dentro do sistema;
- Modularidade: esta propriedade visa a obtenção de módulos no sistema com interfaces bem definidas e serviços encapsulados, desta forma garante-se a autonomia do módulo, mas com mecanismos simples de intercomunicação com outros módulos do sistema;
- Gerenciamento: propriedade que permite o controle, supervisão e coordenação dos elemento e recursos do sistema com o intuito de suportar mudanças nas configurações, a qualidade dos serviços e as regras de negócio impostas pela corporação;

- **Federação:** propriedade que permite a integração de diferentes domínios administrativos e tecnológicos dentro de um sistema distribuído;
- **Segurança:** propriedade que assegura a utilização dos recursos e da informação no sistema, implantando esquemas e mecanismos de segurança. Os requisitos de segurança são diferenciados nos diferentes domínios do sistema e são difíceis de atingir quando existem usuários conectados remotamente e quando recursos e usuários possuem mobilidade dinâmica no sistema;
- **Transparência:** esta propriedade é definida como a capacidade de mascarar ou ocultar aos usuários e às aplicações detalhes e diferenças dos mecanismos utilizados para implementar diversos aspectos relacionados com a distribuição, como por exemplo a localização, migração, replicação entre outros;
- **Proporcionar qualidade do serviço:** esta propriedade visa implantar diversos requisitos da qualidade no comportamento do sistema distribuído.

Num projeto de sistema distribuído e aberto, estas propriedades transformam-se em requisitos do sistema.

2.1.3 Situação Atual da Padronização

O processo de padronização de diferentes aspectos computacionais nos sistemas distribuídos tem sido tarefa árdua, ora pelo processo de padronização de parte das instituições, ora pela aceitação do mercado, mas cujos resultados demonstram que é uma das soluções efetivas na integração de sistemas de informação [Kilov, 93].

O padrão ODP é ciente desta realidade e propõe a padronização do processo de especificação dos sistemas distribuídos e abertos, sendo o nível de

padronização de alta abstração, isto é, define interfaces e componentes abstratos deixando aberta a forma de implementação, protegendo a criatividade e inovação de tecnologia.

No contexto da padronização de sistemas distribuídos e abertos, tem-se outros esforços importantes, como o DCE (Distributed Computing Environment) do grupo OSF (Open System Foundation) e o CORBA (Computer Object Request Broker Architecture) do grupo OMG (Object Management Group), estas propostas possuem objetivos comuns com o padrão ODP [OMG, 95], [Rosemberry, 92].

O padrão ODP visa contribuir e complementar estes esforços, inclusive realizando trabalhos conjuntos com o grupo CORBA, como acontece na padronização da função ODP denominada de "Trader" [Macartney, 92]

Mas é bom ressaltar que a proposta ODP é mais abrangente que as propostas destes grupos, como pode-se analisar nos objetivos das propostas CORBA e DCE que se encaixam dentro contexto do modelo da computação e da engenharia da especificação ODP, sendo estes modelo apenas parte do padrão ODP [Wolisz, 93], [Eckert, 96].

2.1.4 Normas ODP

A normalização ODP oferece a estrutura básica para a criação de diversos tipos de padrões em diferentes formas, seja realizando a padronização de funções ODP ou definindo estratégias de integração de componentes ODP.

Esta normalização abrange também a criação de modelos específicos em domínios específicos, como por exemplo, um modelo ODP para um sistema TMN (Telecommunication Management Network), e também abrange a padronização das funções específicas dentro destes domínios específicos,

como por exemplo, a especificação ODP da função de mediação (Mediation Funtion) dentro do TMN [ITU, 96], [ITU, 92].

Para atingir estes objetivos, a estrutura da normalização ODP está organizada em quatro documentos ou normas que são apresentados a seguir [ISO, 95a], [ISO, 95b], [ISO, 95c], [ISO, 95, d]:

- Recomendação ITU-T X.901 | ISO/IEC 10746-1: este documento denominado de apresentação, contém a apresentação geral do ODP, onde são mostrados os objetivos, justificativas, conceitos básicos e uma revisão geral da arquitetura ODP. Neste documento também são apresentados informações referentes à interpretação e utilização do modelo de referência ODP. Este documento não representa uma norma;
- Recomendação ITU-T X.902 | ISO/IEC 10746-2: este documento denominado de fundamentos, contém os conceitos gerais e estruturas básicas necessárias para poder descrever de forma normalizada qualquer sistema distribuído, assim como também, da o suporte conceitual para poder entender e aplicar a norma ITU-T X.903 | ISO/IEC 10746-3. Este documento é uma norma;
- Recomendação ITU-T X.903 | ISO/IEC 10746-3: este documento denominado de arquitetura, contém as informações relacionadas com a definição das especificações dos cinco pontos de vista, com as funções ODP e com as transparências a implementar. Estas informações contém as características que qualificam o sistema distribuído como aberto. Este documento é uma norma;
- Recomendação ITU-T X.904 | ISO/IEC 10746-4: este documento denominado de semântica arquitetural, contém as informações sobre a formalização dos conceitos básicos de modelagem definidos na norma anterior. Esta formalização é realizada pela interpretação de cada conceito

utilizando diferentes técnicas de descrição formal, como Lotos, Estelle, SDL e Z. Este documento é uma norma.

Recomenda-se a utilização destes documentos nos projetos de sistemas distribuídos e abertos, da seguinte forma: o terceiro documento é o mais importante a utilizar, o segundo dá subsídios teóricos ao terceiro, e o primeiro é o documento de informação geral. O quarto documento deve ser utilizado quando o sistema é de tempo real ou em sistemas que contém requisitos críticos [Kilov, 93], [Li, 95], [Awad, 96].

2.2 Fundamentos ODP

Os fundamentos ODP são conjuntos de conceitos necessários para poder entender e aplicar o modelo de referência ODP. Estes conceitos estão organizados em várias categorias: conceitos primários, conceitos para modelagem, conceitos para especificação, conceitos para arquitetura, e conceitos para conformidade. A figura 2.1 apresenta os conceitos do fundamento ODP.

A seguir serão, apresentados de forma concreta os conceitos mais importantes, mas também serão indicadas as referências bibliográficas onde o leitor poderá obter maiores informações.

2.2.1 Conceitos Primários

O primeiro grupo de conceitos primários é denominado de conceitos de interpretação e englobam um conjunto de meta-conceitos, isto é, conceitos que podem ser aplicados em qualquer tipo de modelagem. Os conceitos deste grupo são [Booch, 94], [UML, 97], [Tanenbaum, 95]:

- Entidade: qualquer coisa real ou abstrata dentro do domínio do sistema que está sendo modelado;

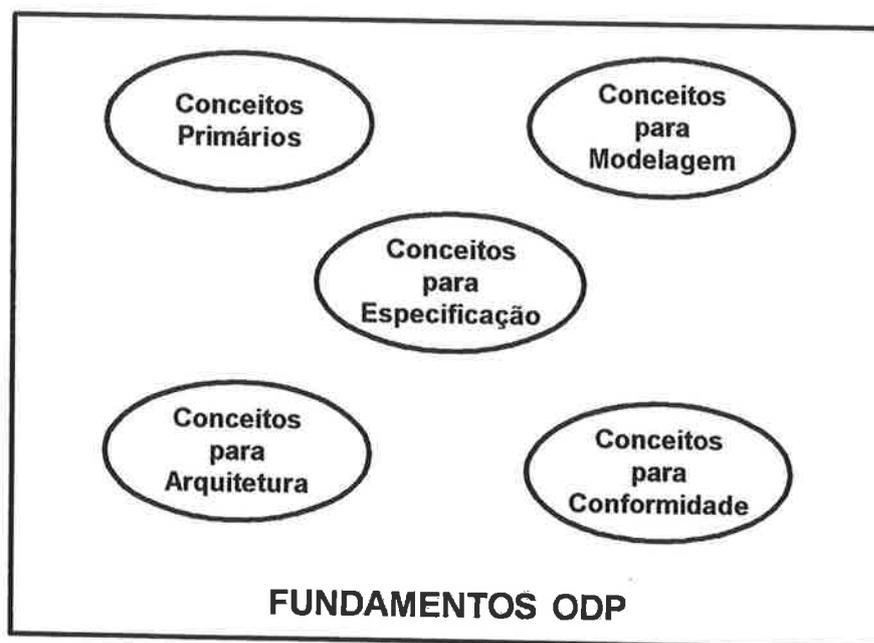


Figura 2.1 Categorias de Conceitos ODP

- **Abstração:** é o processo que retira os detalhes irrelevantes para estabelecer um modelo simplificado;
- **Atômico:** uma entidade é atômica num determinado nível de abstração, quando não pode dividir-se nesse nível de abstração;
- **Arquitetura:** conjunto de regras que definem a estrutura de um sistema assim como as inter-relações entre os elementos que o compõem.

Entre os conceitos primários existe um grupo denominado de conceitos lingüísticos, estes relacionam a sintaxes de qualquer linguagem de especificação ODP com os conceitos de interpretação. Existem dois conceitos:

- **Termo:** construção lingüística que pode ser usada para representar uma entidade;
- **Sentença:** construção lingüística que contém um ou mais termos e que apresenta o relacionamento de várias entidades.

2.2.2 Conceitos para Modelagem

Esta categoria de conceitos é orientada para construção de arquiteturas ODP, isto significa que os modelos obtidos nas arquiteturas devem utilizar estes conceitos. Os principais conceitos são:

- **Objeto:** é um modelo de uma entidade, que é definido por seu comportamento e pelos pontos de interação com o ambiente externo;
- **Comportamento:** o comportamento de um objeto é definido como um conjunto de ações e estados;

- Interface: é uma abstração do comportamento do objeto que consiste de um subconjunto das interações. Cada interface do objeto é identificada com um conjunto de interações com outros objetos.

Cabe ressaltar que existem outros conceitos envolvidos como pode-se ver em [Rumbaugh, 91], mas os conceitos desta categoria formam parte da conceituação do paradigma de orientação a objetos, portanto, as referências bibliográfica indicadas a seguir apresentam estes conceitos com maior detalhe [Hirama, 95], [Booch, 94], [Larman, 97], [Coad, 91].

2.2.3 Conceitos para Especificação

Esta categoria de conceitos está relacionada com os requisitos que devem ser considerados nas linguagens utilizadas na especificação de sistemas ODP. Os principais conceitos são:

- Composição: a integração de dois ou mais objetos com o objetivo de compor um outro objeto em um nível de abstração diferente. As interações entre os objetos integrados são encapsuladas pelo objeto resultado;
- Compatibilidade comportamental: um objeto é compatível com outro objeto, quando o primeiro objeto pode substituir o segundo objeto sem produzir modificações ou interferências no ambiente externo;
- Tipos e classes: um tipo é um predicado que caracteriza a um conjunto de elementos; este elementos são englobados dentro de uma classe;
- Template: é a especificação detalhada das características de uma classe;
- Hierarquia de classes: estrutura hierárquica de classes, onde existe uma classe superior denominada de superclasse e classes que são denominadas de subclasses;

- Instanciar (instantation): procedimento que implica a produção de um objeto a partir de um template determinado.

Maiores detalhes sobre estes e outros conceitos referentes a orientação a objetos podem ser encontrados em [UML, 97], [Hirama, 95].

2.2.4 Conceitos para Arquitetura

São os conceitos que estão relacionados com diversos aspectos dos sistemas distribuídos e da distribuição computacional, e que podem ou não estar suportados pelas linguagens de especificação das arquiteturas ODP. Os principais conceitos são:

- Grupo e domínio: os grupos são conjuntos de objetos agrupados por alguma relação particular que pode ser uma relação estrutural entre objetos ou porque seus comportamentos possuem características em comum. Estes grupos ou grupo podem ser controlados por um objeto autoridade, neste caso é denominado de domínio;
- Contrato: acordos que governam parte do comportamento de um conjunto de objetos. Nestes acordos especificam-se as obrigações, permissões e proibições dos objetos envolvidos;
- Qualidade do serviço: um conjunto de requisitos de qualidade a ser implantado no comportamento dos objetos. Estes requisitos devem ser especificados no contrato, medidos e armazenados em relatórios;
- Nome: é um termo colocado a uma entidade dentro de um contexto definido, este identificador servirá para que o objeto seja reconhecido e acessado dentro do sistema distribuído;

- Conexão (binding): definição abstrata de um contexto onde duas o mais interfaces de objetos estão envolvidas e interagem segundo um contrato comportamental previamente definido.

O detalhamento destes e outros conceitos definidos na norma podem ser encontrados em [ISO, 95b], [Shaw, 96].

2.2.5 Conceitos Referentes à Conformidade

A conformidade é um tópico de grande interesse na padronização ODP, existe a preocupação de apresentar os processos e mecanismos que permitam demonstrar que a implementação do sistema deve estar conforme com a especificação [ISO, 94]. Dentro deste contexto o padrão ODP apresenta os conceitos de conformidade e concordância.

A conformidade é a relação que existe entre a especificação e a implementação, isto é, o que foi definido na especificação deve ser mapeado diretamente na implementação.

A concordância é a relação entre duas especificações, isto é, duas especificações estão em concordância quando os requisitos definidos numa das especificações são atingidos totalmente pela outra especificação.

A conformidade é analisada através de pontos de conformidade ou de pontos de referência, estas interfaces permitem a realização de diversos testes para avaliar o comportamento, e cujos resultados permitem ver os graus de conformidade com a especificação. Existem vários tipos de pontos de referência definidos no padrão ODP, que a seguir são apresentados:

- Ponto de referência programado: é uma interface programada que deve ser implantada para ter acesso a uma função determinada, e cujos requisitos de

conformidade estão definidos segundo o contexto da compatibilidade comportamental entre os objetos;

- Ponto de referência perceptível: é uma interface que permite a interação entre o sistema e o mundo externo, e cujos requisitos de conformidade estão definidos em termos da informação apresentada aos usuários. Exemplo típico são as interfaces homem-máquina;
- Ponto de referência de intercomunicação: é uma interface que permite a comunicação entre vários sistemas, e cujos requisitos de conformidade estão definidos em termo da informação trocada entre os sistemas;
- Ponto de referência de armazenamento: é uma interface que permite a conexão de um elemento armazenador externo com o sistema, e cujos requisitos de conformidade estão definidos em termos do aspecto comportamental do armazenador externo, desta forma permite-se gravar informações em um sistema para depois ser utilizado em outro sistema.

O modelo de referência ODP identifica na arquitetura da especificação, certos pontos de referência que se tornam como possíveis candidatos a pontos de conformidade.

2.3 Os Pontos de Vista

A estrutura principal do modelo de referência ODP é denominada de arquitetura da especificação ODP e está constituída de duas partes: os pontos de vistas e as funções ODP. Neste item define-se o termo ponto de vista, assim como, apresenta-se a estrutura dos cinco pontos de vista: empresa, informação, engenharia, computação e tecnologia [Joyner, 97].

2.3.1 Definição

O conceito de ponto de vista aparece com o objetivo de manipular as informações geradas na especificação dos sistemas distribuídos. Estas informações englobam vários aspectos de projeto como por exemplo: comunicação, transparência, segurança, entre outros, cujas definições não caberiam numa descrição simples [Palme, 92], [Herbert, 94], [Geihs, 93].

Este problema foi umas das motivações do padrão ODP, e como consequência disto, foi proposta a definição do termo ponto de vista, a identificação dos cinco pontos de vistas básicos e a estruturação de uma linguagem de especificação para cada ponto de vista.

Um ponto de vista pode ser entendido de diferentes aspectos, que são apresentado a seguir:

- **Nível de abstração:** um ponto de vista representa uma abstração do sistema, isto é, uma forma de definir ou de ver o sistema no desenvolvimento. Em especial o conjunto de cinco pontos de vista do padrão ODP representa o nível de abstração da fase de especificação do projeto;
- **Ferramenta de projeto:** na norma ODP, cada ponto de vista está associado a uma determinada linguagem de especificação, que na verdade, é um conjunto de conceitos, regras e procedimentos que permitem a estruturação do ponto de vista em linguagem natural e em modelos de objetos. Significa que um ponto de vista é uma ferramenta de projeto para modelar a especificação do sistema;
- **Ciclo de vida do projeto:** os pontos de vista ou um ponto de vista representam a fase de especificação do sistema, estes pontos de vistas

possuem requisitos específicos que deverão ser atingidos na implementação do sistema;

- Organização: um ponto de vista constitui um conjunto de informações (repositórios) referentes a uma área de interesse no projeto de um sistema distribuído, isto é, representa uma subdivisão da especificação do sistema.

Neste sentido, o objetivo destas sub-especificações é organizar a complexidade inerente das informações dos requisitos do sistema, assegurando a manipulação consistente desta informação ao longo do projeto.

2.3.2 Formas de Utilização

O conjunto de especificações referentes aos pontos de vista definidos pelo padrão ODP podem ser utilizados de diferentes formas no desenvolvimento de projetos de sistemas distribuídos.

É bom lembrar que a especificação do ponto de vista está constituída de informações em linguagem natural (texto) e em conjuntos de diagramas orientados a objetos.

Nesse sentido, não existe uma estratégia formal de utilização, como a norma ressalta os pontos de vista não representam as fases de uma metodologia nem muito menos uma arquitetura de sistema tipo OSI (Open System Interconnection), portanto recomenda que a estratégia de utilização deve se adequar ao tipo de sistema que se pretende projetar, [Sinderen, 95].

Entretanto, uma estratégia de utilização deve considerar duas partes: a definição do esquema de prioridades e a definição do tipo de informações a manipular.

O esquema de prioridades diz respeito à ordem de importância entre os pontos de vista, isto é, segundo um determinado contexto são colocados graus de prioridade em cada ponto de vista.

Por exemplo, no projeto de um sistema de comunicação de rede onde a fase atual é a definição da arquitetura, um esquema de prioridades possível é o seguinte: engenharia, computação, informação, tecnológico, empresa.

O outro aspecto é a definição do tipo de informação, que diz respeito a utilização da informação, ora documento em linguagem natural ora diagramas de objetos, ou os dois. Por exemplo, se o profissional envolvido numa determinada tarefa possui fortes conhecimentos sobre orientação a objetos, convém neste caso utilizar os diagramas de objetos como fonte de especificação.

Para definir esta estratégia devem ser considerados ou analisados os seguintes critérios: a natureza ou escopo do sistema, isto é, as características gerais do sistema ou do domínio da aplicação, o contexto definido pela fase do desenvolvimento específico, e, finalmente, o tipo de profissional que vai manipular a especificação. A tabela 2.1 apresenta exemplos utilizando a estratégia proposta.

Um exemplo real sobre a utilização dos pontos de vista, encontra-se nos projetos de sistemas TMN (Telecommunication Management Network) cuja estruturação está sendo padronizada pela ITU na norma que está na referência [ITU, 96].

Nesta norma sugere-se a utilização de apenas quatro pontos de vista, deixando de lado o ponto de vista tecnológico cuja conceituação não está dentro do escopo do setor de telecomunicações da ITU, devido a posição da instituição de fornecer propostas independentes da tecnologia. Neste caso o escopo da área de atuação definiu a utilização.

Membros da Equipe	Fases do Desenvolvimento	Escopo do Sistema	Conhecimentos do Profissional	Esquema de Prioridades	Documentos a Utilizar
Analista de Sistemas	Análise	Sistema de Informação para Automação Bancária	Possui conhecimentos OO	Informação Computação Engenharia Empresa Tecnológico	Diagrama de objetos
Engenheiro	Projeto	Sistemas de Redes	Possui conhecimentos OO	Engenharia Computação Informação Tecnologia Empresa	Diagrama de objetos
Programador	Implementação	Sistemas de Interface Homem – Máquina	Não possui conhecimentos OO	Tecnologia Informação Computação Engenharia Empresa	Documento técnico
Líder Técnico	Todas	Sistemas de Informação para Telecomunicações	Possui conhecimentos OO	Engenharia Computação Tecnologia Informação Empresa	Documentos técnicos e diagramas de objetos
Gerente de Projeto	Todas	Sistemas de Informação para Manufatura	Não possui conhecimentos OO	Empresa Informação Tecnologia Computação Engenharia	Documentos técnicos

Tabela 2.1 Exemplos de Formas de Utilização dos Pontos de Vista

2.3.3 Ponto de Vista da Empresa

O ponto de vista da empresa define os requisitos básicos do sistema considerando o contexto da empresa onde deverá funcionar.

Neste ponto de vista são definidos o objetivo e escopo da empresa, as políticas que define a atuação do sistema, e as políticas que definem o inter-relacionamento da empresa com entidades externas.

A estrutura básica do ponto de vista da empresa contém os seguintes aspectos:

- Escopo da empresa: contém as características principais da empresa enxergando a empresa como uma entidade de negócios, este ponto de vista engloba todos os tipos de empresa;
- Funções empresariais: funções realizadas por os elementos ou objetos dentro da empresa, estas funções são regulamentadas pelas políticas da empresa. Exemplo de funções empresariais são: o presidente, o administrador, o fornecedor de serviços, o vendedor, o gerente entre outros elementos;
- Políticas: são regras definidas na empresa internamente ou definidas por agentes externos à empresa, normalmente estão constituídas de obrigações, permissões e proibições. Um exemplo de política pode-se encontrar na área bancária onde se define regras para determinar os benefícios e obrigações do banco e do cliente na negociação de um empréstimo de dinheiro;
- Procedimentos: contém um conjunto de ações específicas realizadas pelos os objetos empresa. Um exemplo deste aspecto é: o procedimento de

inclusão de um equipamento dentro de uma célula flexível de manufatura, [Becerra, 94];

- Comunidade: constituída de grupos de objetos empresa ou de funções empresariais que realizam suas atividades dentro de política definidas e que visam um objetivo comum. Um exemplo de uma comunidade dentro de um sistema de informação hospitalar está representado pelo contexto definido dentro do serviço de atendimento ambulatorio;
- Federação: constituída de várias especificações de empresas, ou sub-empresas ou domínios e cuja estrutura deve definir o relacionamento entre estes elementos. Um exemplo de federação é o sistema de informação corporativo de uma empresa que possui subsidiárias em várias partes do mundo;

A figura 2.2 apresenta o núcleo do ponto de vista da empresa onde são mostrados os seis elementos supracitados.

2.3.4 Ponto de Vista da Informação

O ponto de vista da informação define a semântica da informação e a semântica do processamento da informação no sistema que deverá estar constituído de objetos informação.

A estrutura do ponto de vista da informação ou a semântica da informação é definida dentro de três tipos de esquemas que a seguir são apresentados:

- Esquema invariante: conjunto de condições ou predicados que sempre são verdadeiros e que se aplicam a um ou a um conjunto de objetos informação.

Estas condições regulamentam os possíveis estados ou mudanças de estados dos objetos informação. Um exemplo deste tipo de esquema

- encontra-se num sistema de informação comercial de vendas utilizando tecnologia Web e Internet, quando se define no objeto informação, denominado de pedido de venda, uma condição que determina a aprovação automática para um valor de vendas menor que quinze mil reais;
- Esquema estático: é a especificação de um estado de um objeto ou conjunto de objetos informação, em um ponto específico dentro do tempo de processamento. A definição do esquema estático está limitado pelo esquema invariante e constitui um sub-tipo deste esquema. Utilizando o mesmo caso apresentado no esquema anterior, um exemplo deste esquema estático é o valor inicial de um objeto informação para ser considerado pedido de vendas que deverá ser de dois mil reais;
 - Esquema dinâmico: define todos os estados de um objeto ou conjunto de objetos informação dentro do tempo de processamento. O esquema dinâmico está governado pelo esquema invariante e se constitui um sub-tipo deste esquema. Exemplo do esquema dinâmico, são os descontos automáticos aplicados no objeto informação, do exemplo anterior, quando atinge os valores de cinco mil, dez mil e quinze mil reais.

A figura 2.3, apresenta o núcleo do ponto de vista da informação ressaltando a relação entre os esquemas definidos.

2.3.5 Ponto de Vista da Computação

O ponto de vista da computação define o sistema como uma estrutura de processadores de informação distribuídos que interagem através de interfaces específicas. A figura 2.4 apresenta a estrutura do ponto de vista computacional.

Neste ponto de vista existe a preocupação com a distribuição das aplicações dentro do sistema, sem se preocupar com a infra-estrutura de comunicação. A figura 2.5 mostra esta transparência.

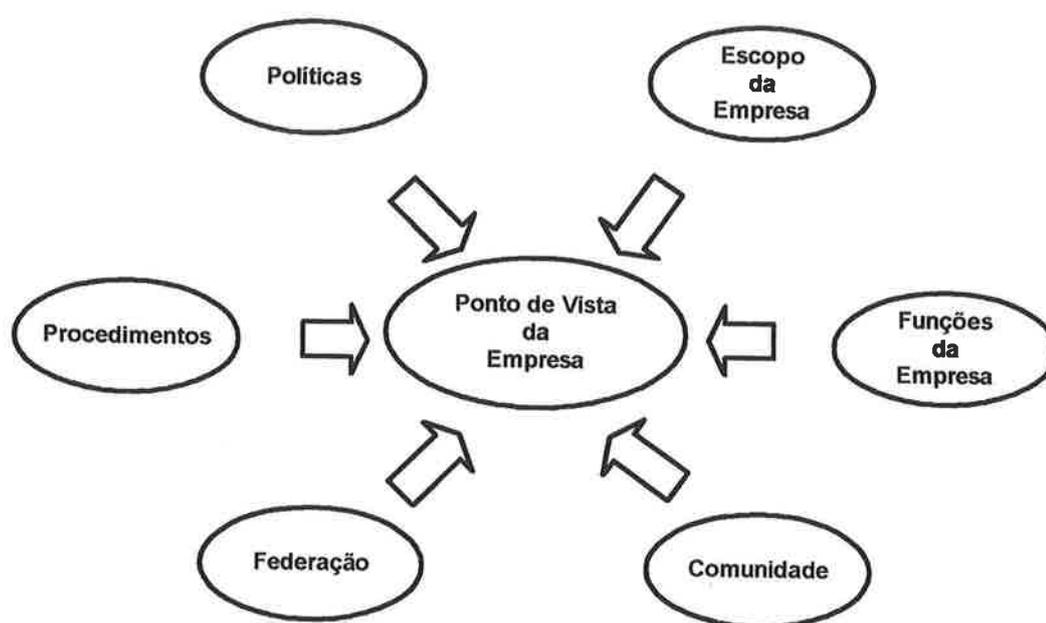


Figura 2.2 Estrutura do Ponto de Vista da Empresa



Figura 2.3 Estrutura do Ponto de Vista da Informação

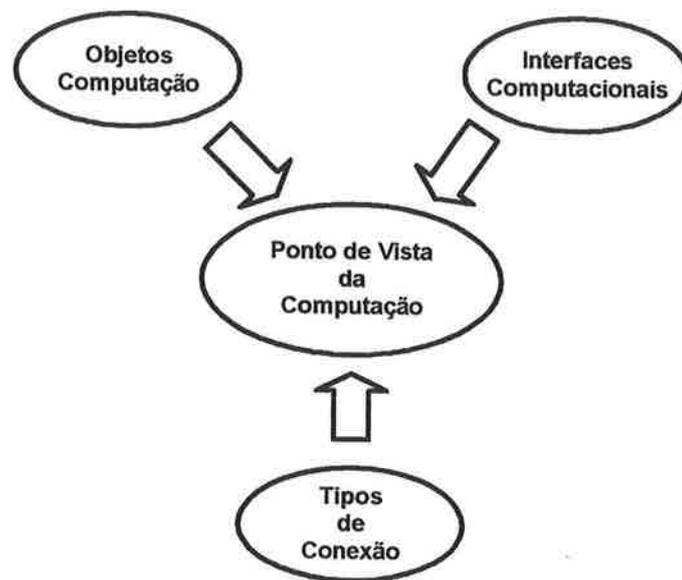


Figura 2.4 Estrutura do Ponto de Vista da Computação

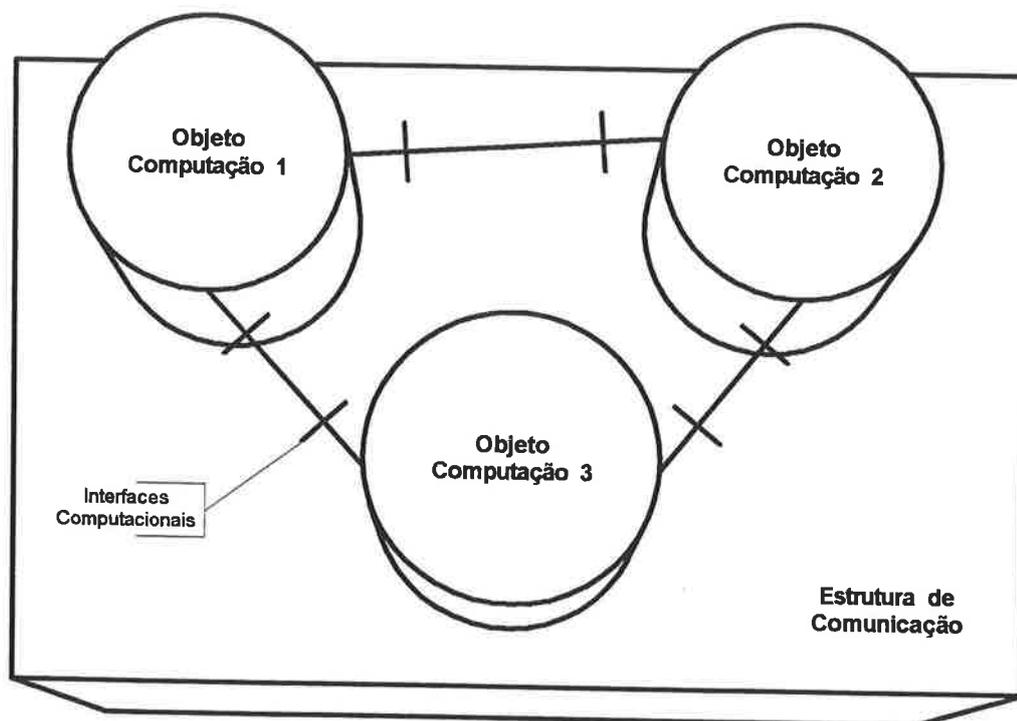


Figura 2.5 Relação Entre os Objetos Computação e a Estrutura de Comunicação

Dentro do escopo da visão da computação, os processadores de informação ou aplicações são representados como objetos computação, que são interligados através de interfaces computacionais estruturando um sistema distribuído.

Um objeto computação encapsula suas atividades e estados, sendo estes estados acessados e modificados através das operações executadas nas interfaces computacionais.

Desta forma o sistema possui graus de paralelismo, pois são executadas diversas tarefas concorrentes nos diferentes objetos computacionais. Também, devido à característica de encapsulamento, um o objeto computacional pode representar uma aplicação não distribuída, desta forma podem-se reutilizar os recursos desenvolvidos dentro de uma visão centraliza [Nicol, 93].

Os elementos do ponto de vista da computação são os objetos computacionais, as interfaces computacionais e os tipos de conexão, que a seguir serão apresentados

2.3.5.1 Objeto Computação

Um objeto computação é o componente básico do modelo do ponto de vista da computação, e representa os aplicativos distribuídos. Este objeto computação é definido por uma especificação geral ou molde (template) e possui os seguintes elementos:

- Interface computacional: definido por uma especificação geral (template) da interface, onde são identificados as formas de interação, os comportamentos das interfaces e os contratos de interação externa;

- Especificação do comportamento: define-se os estados e serviços executados pelos objetos computação;
- Requisitos externos: restrições colocadas pelo ambiente externo, normalmente definidas em contratos.

2.3.5.2 Interfaces Computacionais

Uma interface computacional é um elemento que é utilizado pelos objetos computação para realizar seus pedidos de serviço. O tipo de pedido de serviço ou interação entre objetos determina três formas de interface: interface para operações, interfaces para fluxos e interfaces para sinais.

- Interface para operações: é a interface entre objetos computação, neste contexto definidos como servidor e cliente, e suporta interações denominadas operações que são estruturas semelhantes com os procedimentos ou com as chamadas de rotinas utilizados na programação de sistemas.

Existem dois tipos de operações: o tipo interrogação onde o servidor retorna uma resposta para o cliente, e o tipo anúncio onde não existe resposta para o cliente. A estrutura de uma operação é definida por uma assinatura que está composta do nome da operação, o nome do objeto segundo a função exercida (servidor ou cliente), os parâmetros e as formas de finalizar a interação;

- Interface para fluxos: é a interface entre objetos computação, neste contexto definida como produtor e consumidor, e suporta interações denominadas de fluxos, que representam fluxos contínuos de informação como o fluxo de áudio e vídeo numa aplicação multimídia [Coulson, 95]. Uma interação tipo fluxo é definida por um assinatura que compõe um

nome, a identificação do produtor ou consumidor, e o tipo de informação transmitida;

- Interfaces para sinais: é a interface entre objetos computação, neste contexto definidos como iniciador e respondedor, e suporta interações denominadas de sinais, que representam invocações como eventos e interrupções. A estrutura do sinal está constituída do nome, a identificação do iniciador ou respondedor, e os parâmetros. Cabe ressaltar que as operações e os fluxos podem ser definidos em termos de sinais.

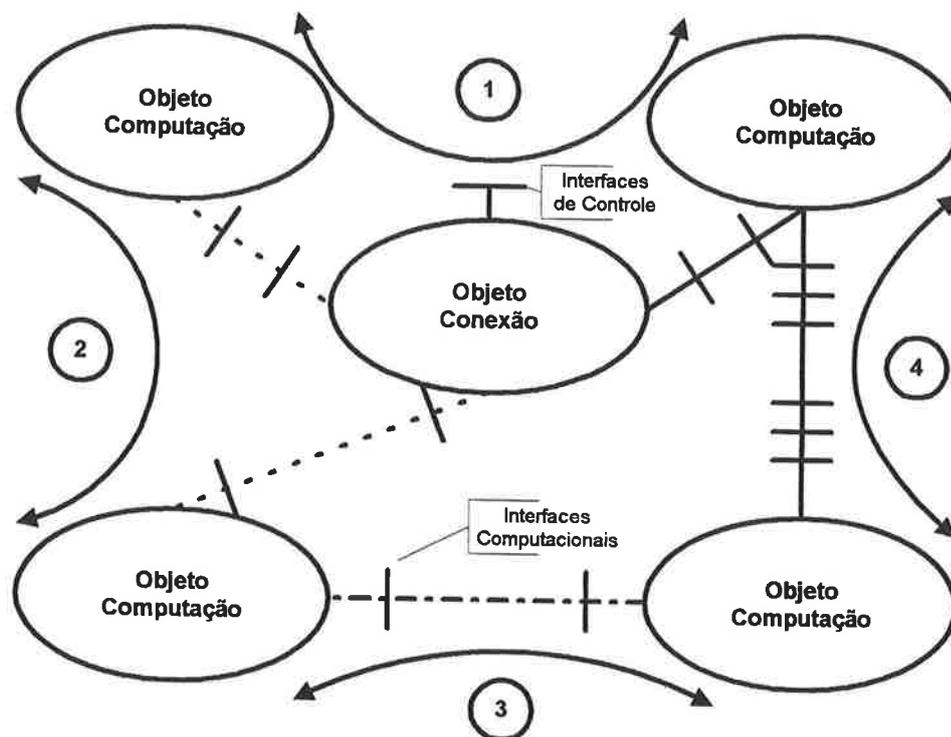
2.3.5.3 Tipos de Conexão

A interação entre dois objetos computação é realizada entre interfaces, que devem ser conectadas através de uma via de comunicação. O processo de estabelecer uma ligação entre duas interfaces é denominada de conexão, no modelo ODP existem dois tipos de conexão: explícito e implícito.

A conexão explícita é configurada quando existem interações entre dois objetos computação cujas interfaces manipulam operações e fluxos contínuos de informação.

Alem disso, uma conexão explícita é denominada primitiva quando se conectam dois objetos computação que possuem interfaces do mesmo tipo e cujas estruturas são compatíveis. E uma conexão explícita é composta quando se conectam duas interfaces de diferentes tipos ou iguais através de um objeto conexão.

A conexão implícita é configurada entre interfaces para operação, e especialmente utilizada em linguagens procedurais onde não precisa explicitar a ação de conexão. A figura 2.6 mostra os tipos de conexão em um modelo de computação.



Legenda

- 1 Conexão explícita composta com interfaces diferentes
- 2 Conexão explícita composta com interfaces iguais
- 3 Conexão explícita primitiva
- 4 Conexão implícita entre interfaces para operação

Figura 2.6 Tipos de Conexão Computacional

2.3.6 Ponto de Vista da Engenharia

O ponto de vista da engenharia preocupa-se por suportar a interação entre objetos, desta forma especifica mecanismos, funções e estruturas que viabilizam a comunicação entre objetos computação distribuídos no sistema ODP.

O escopo do ponto de vista da engenharia será apresentado dentro de três elementos fundamentais: o objeto básico de engenharia, a estrutura gerencial e a estrutura de comunicação. A figura 2.7 apresenta o núcleo do ponto de vista da engenharia.

2.3.6.1 Objeto Básico de Engenharia

No contexto definido pelo ponto de vista da engenharia, o objeto computação definido no modelo de computação transforma-se em objeto básico de engenharia. As atividades de interação entre objetos básicos de engenharia é produto do relacionamento com outros objetos engenharia que possuem funções específicas.

A importância deste objeto básico de engenharia sobre os outros objetos do mesmo modelo, é que este representa o resultado de uma correspondência direta e clara entre o ponto de vista da computação e o ponto de vista da engenharia. Esta correspondência não é definida claramente em outros pontos de vista discutidos em itens anteriores.

2.3.6.2 Estrutura de Gerenciamento

A figura 2.8 apresenta a estrutura de gerenciamento que compõe o ponto de vista de engenharia, o objetivo principal desta estrutura é de organizar a infra-

estrutura do sistema para otimizar a forma de gerenciamento dos objetos distribuídos. Os elementos que constituem esta estrutura são:

- Grupo: representa a configuração de um conjunto de objetos básicos de engenharia de forma a compor um grupo compacto com o propósito de executar funções específicas como ativação, desativação, reativação, recuperação e migração;
- Gerenciador de grupo: é um objeto de engenharia que gerencia os objetos básicos correspondente a seu grupo;
- Cápsula: corresponde a uma configuração de objetos de engenharia de forma a criar uma estrutura simples e compacta com o objetivo de encapsular o processo e o armazenamento;
- Gerenciador de cápsula: é um objeto de engenharia que realiza o gerenciamento dos objetos de engenharia da respectiva cápsula;
- Nó: representa a configuração de objetos de engenharia de forma a compor um processador de informação e onde são executadas as funções de processamento, armazenamento e comunicação, este processador possui uma identificação que permite sua localização no espaço;
- Núcleo: é um objeto de engenharia encarregado da coordenação das funções de processamento, armazenamento e comunicação dentro do nó.

O relacionamento entre objetos gerenciadores é realizado através das interfaces de gerenciamento previamente configuradas, mostradas como interfaces de controle na figura 2.8. Cabe ressaltar que o padrão ODP não formaliza a estrutura de comunicação entre nó, cápsula e grupo.



Figura 2.7 Estrutura do Ponto de Vista da Engenharia

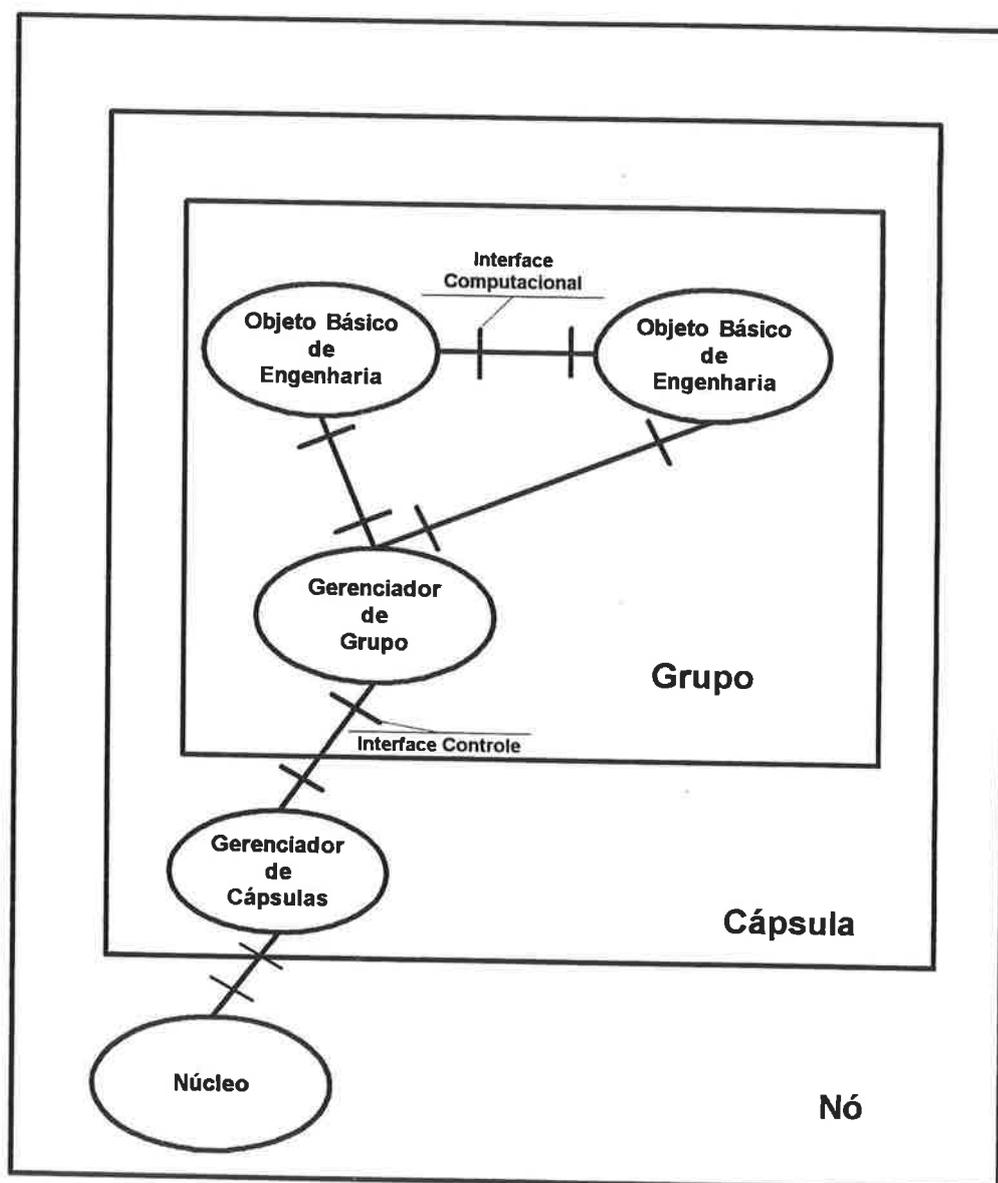


Figura 2.8 Estrutura Gerencial do Ponto de Vista da Engenharia

2.3.6.3 Estrutura de Comunicação

A comunicação entre objetos de engenharia pode ser realizado em duas formas, quando os objetos pertencem ao mesmo grupo ou nó utiliza-se uma conexão e pode se implantar utilizando mecanismos específicos do sistema, esta forma não é padronizada; e quando os objetos pertencem a diferentes cápsulas ou nós utiliza-se uma conexão distribuída, cuja estrutura é padronizada no ODP.

Para implantar esta conexão distribuída é necessário configurar uma estrutura denominada de canal, cujo objetivo é de oferecer de forma transparente os serviços de comunicação aos objetos que estão interagindo. O canal é constituído dos seguintes objetos de engenharia:

- Adaptador: é um objeto que interage diretamente com o objeto básico de engenharia, oferecendo os serviços de conversão dos dados que são gerados na interação. As interfaces que o conectam com o objeto básico e com o conector, possuem uma interface de controle para interatuar com um objeto gerenciador;
- Conector: é um objeto encarregado de gerenciar a integridade ponto-a-ponto do canal, supervisionando as falhas nos objetos e na comunicação e, se fosse o caso, executa mudanças na configuração e na comunicação. Possui uma interface de controle;
- Protocolo: este objeto é o encarregado de oferecer as funções de comunicação. Possui uma interface de controle utilizada para gerenciamento;

- **Interceptor:** este objeto é configurado quando o canal é utilizado para comunicar dois domínios diferentes, e é o encarregado de realizar a conversão de protocolos.

A figura 2.9 apresenta a estrutura do canal, neste exemplo o canal dá suporte a dois objetos computação tipo cliente e servidor.

2.3.7 Ponto de Vista da Tecnologia

Este ponto de vista preocupa-se com a seleção de tecnologia para implementar o sistema ODP. A figura 2.10 apresenta o núcleo do ponto de vista da tecnologia, nesta figura os elementos são apenas exemplos, estes dependem do sistema a implementar.

Neste contexto deve-se identificar os recursos de hardware ou software que serão utilizados na implementação dos objetos, este recursos podem ser padronizados ou proprietários ou ambos, dependendo da política da empresa [ITU, 96].

Apesar da especificação ter que ser independente da tecnologia, a utilização deste ponto de vista deve ser analisada de forma estratégica, principalmente analisando o tipo de projeto a desenvolver.

Existem projetos na área da automação, como usinas nucleares, onde não pode-se especificar sem considerar a tecnologia, e existem projetos como o desenvolvimento na área de telecomunicações que não se deve utilizar o aspecto tecnologia [ISO, 96].

Finalmente, o modelo do ponto de vista da tecnologia está constituído de objetos tecnologia que possuem uma correspondência direta com os objetos do ponto de vista de engenharia.

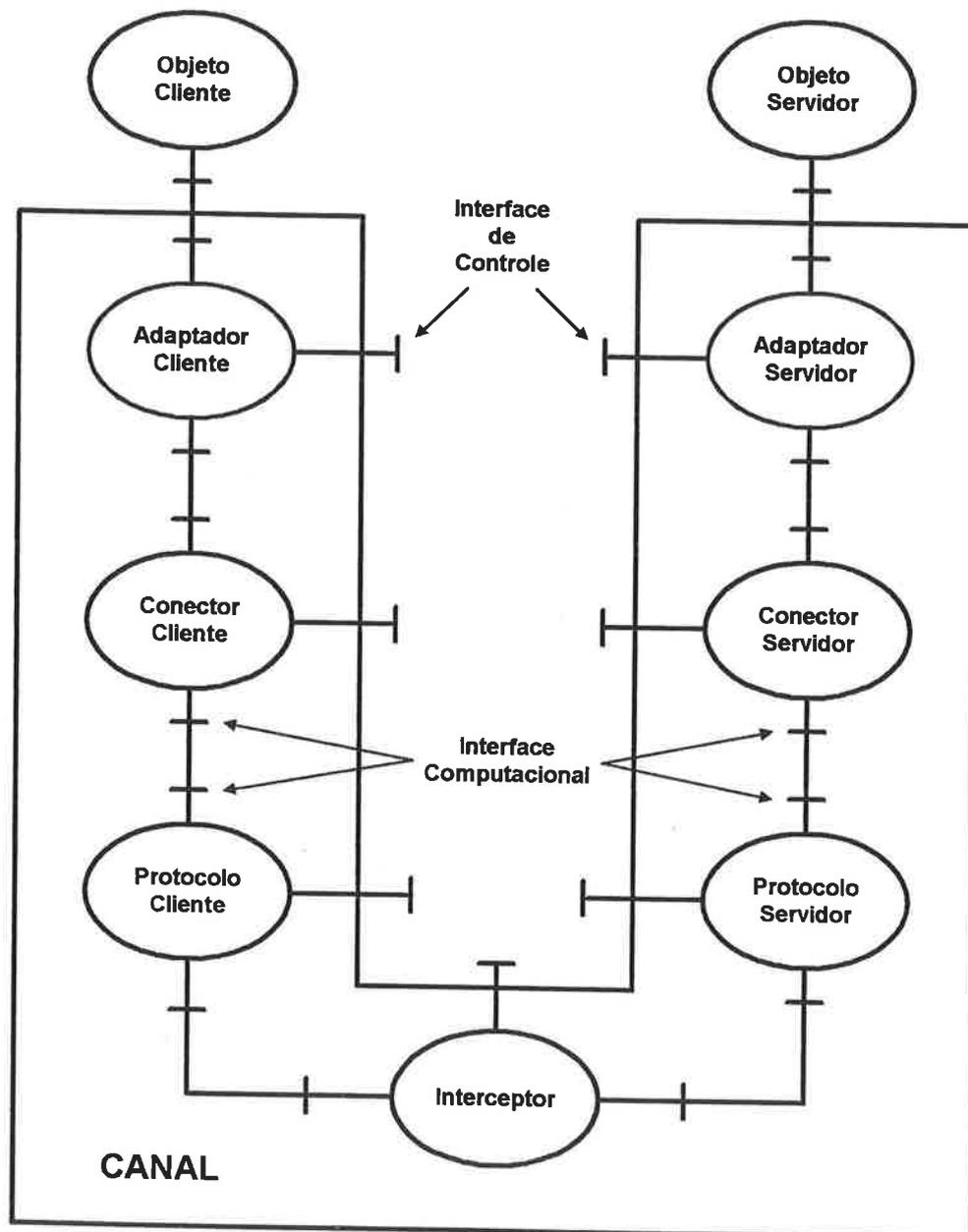


Figura 2.9 Estrutura da Comunicação do Ponto de Vista da Engenharia

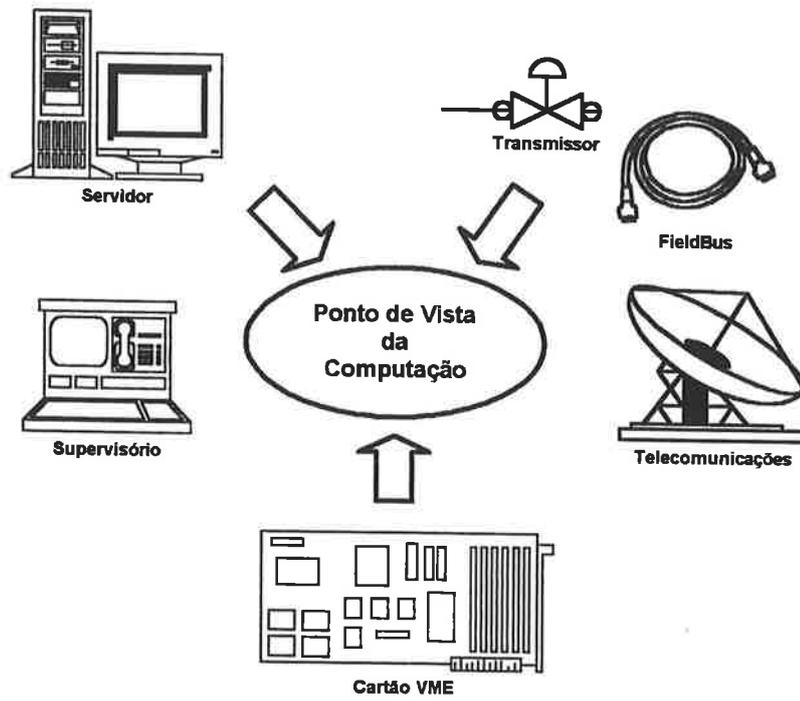


Figura 2.10 Estrutura do Ponto de Vista da Tecnologia

2.4 Transparências e Funções

A especificação baseada em pontos de vista identifica elementos e estruturas do sistema; completando esta especificação, existem outros requisitos em termos de integração e cooperação que o padrão ODP define através de dois conceitos: transparências e as funções ODP, que a seguir serão analisados.

2.4.1 Transparências Distribuídas

As transparências distribuídas são mecanismos abstratos cujo objetivo é esconder aos sistemas e usuários as complexidades dos serviços oferecidos pela estrutura distribuída, otimizando a execução de tarefas cooperativas entre os elementos do sistema.

Um operador utiliza um sistema supervisorio em um sistema de automação de uma subestação elétrica e observa em tempo real variáveis do processo, ele não se preocupa com a complexidade tecnológica que está atrás da interface homem-máquina. Um exemplo de transparência no nível de usuário.

Um aplicativo Web que traz informação de um servidor SQL não se preocupa com a transformação de protocolos, existe um serviço na infra-estrutura que realiza esta função. Transparência no nível de aplicativo.

Dois objetos computacionais interagindo não se preocupam com a forma de comunicação, existe um canal disponível configurado no nó que suporta esta ação. Transparência no nível de objeto.

O padrão ODP define as seguintes transparências distribuídas:

- **Transparência de acesso:** refere-se aos mecanismos que manipulam as diferenças na representação dos dados e às formas de invocação quando existe a intercomunicação entre dois objetos;
- **Transparência de falha:** refere-se aos mecanismos de tolerância a falha aplicados nos objetos;
- **Transparência de localização:** refere-se ao uso da informação sobre a localização no espaço de um objeto no sistema;
- **Transparência de migração:** refere-se aos mecanismos que permitem a mudança da localização do objeto no sistema;
- **Transparência de persistência:** refere-se aos mecanismos que permitem a ativação e desativação dos objetos nos processos distribuídos;
- **Transparência de replicação:** refere-se aos mecanismos que permitem a manipulação de objetos idênticos como se fosse um único objeto;
- **Transparência de realocação:** refere-se aos mecanismos que manipulam o restabelecimento da conexão com uma interface que realizou uma migração;
- **Transparência transacional:** refere-se aos mecanismos que coordenam as transações que são atividades de configuração de um objeto com o objetivo de manter a consistência em seus estados. As transações utilizam os esquemas invariantes e dinâmicos.

2.4.2 Funções ODP

A implementação de uma transparência será realizada utilizando objetos computacionais denominados de funções ODP. As funções ODP especificam

serviços cujo objetivo é estruturar as transparências distribuídas, nesta tarefa podem ser utilizadas uma ou várias funções ODP ou outras funções específicas do sistema distribuído.

Desta forma o padrão ODP apresenta quatro categorias de funções que a seguir serão apresentadas [ISO, 95c].

2.4.2.1 Funções de Gerenciamento

Correspondem as funções referentes ao gerenciamento segundo a estrutura definida no item 2.3.6.2, estas funções são as seguintes:

- Funções de gerenciamento do nó: corresponde ao controle das funções de processamento, armazenamento e comunicação que são implantadas no nó. Esta função engloba o gerenciamento das linhas de controle dos processos, da criação de canais de engenharia, da localização das interfaces entre objetos engenharia, da criação e desativação de cápsulas, e do gerenciamento dos relógios de sincronização que são configurados dentro do nó;
- Funções de gerenciamento da cápsula: engloba o gerenciamento da criação e desativação dos grupos dentro da cápsula e da desativação de todos os elementos da cápsula;
- Funções de gerenciamento do grupo: abrange o gerenciamento do armazenamento da configuração do grupo, da recuperação do grupo, da reativação do grupo, da migração do grupo, e da desativação dos elementos dentro do grupo;
- Funções de gerenciamento do objeto: engloba o gerenciamento do armazenamento da configuração do objeto e a desativação do objeto;

2.4.2.2 Funções de Coordenação

Este grupo de funções é o encarregado de implementar os serviços que permitem a coordenação das atividades dos objetos engenharia, alias esta funções oferecem suporte à execução das funções de gerenciamento. Estes serviços são os seguintes:

- Função de notificação de eventos: serviço que cuida da estruturação do histórico dos eventos dos objetos;
- Função de controle da configuração: serviço que cuida da estruturação da configuração dos grupos, isto é, armazenamento dos estados dos objetos e interfaces atuais;
- Função de recuperação: serviço que executa os mecanismos de recuperação quando ocorre uma falha no grupo;
- Função de desativação e reativação: serviço que executa os mecanismos de desativação e de reativação dos grupos;
- Função de migração: serviço que executa os mecanismos que coordenam a migração dos grupos entre cápsulas diferentes;
- Função de replicação: serviço que executa mecanismos que permitem a manipulação de um grupo de objetos cujos comportamentos são idênticos;
- Função transacional: serviço que executa mecanismos que asseguram a consistência das transações.

2.4.2.3 Funções dos Repositórios

As funções dos repositórios encarregam-se da manipulação das informações referentes às interfaces, objetos e outros elementos do sistema. Os serviços deste grupo são:

- Função de armazenamento: serviço que permite o armazenamento de informações em repositórios de dados;
- Função de organização da informação: serviço que permite o gerenciamento da organização da informação definidos segundo os esquemas do ponto de vista da informação;
- Função de realocização e tipos: serviço permite o gerenciamento da informação da localização das interfaces e dos tipos de especificações;
- Função de localização de serviços (Trading): serviço que gerencia o repositório de serviços que são oferecidos aos elementos do sistema. Estes serviços podem estar em outros domínios.

2.4.2.4 Funções de Segurança

O grupo destas funções oferece serviços que permitem implantar diferentes níveis de segurança no sistema. Os serviços que fazem parte deste grupo são:

- Função de controle de acesso: controla o acesso não autorizado com objetos do domínio;
- Função de auditoria: permite a monitorização, análise e coleta de informação dos eventos relacionados com a segurança do sistema;

- Função de autenticação: permite validar a identidade dos objetos, confirmando sua configuração do objeto dentro de um domínio específico;
- Função de integridade: detecta e evita a manipulação não autorizada das informações dentro do sistema;
- Função de confidencialidade: evita a divulgação de informação não autorizada do sistema;
- Função de gerenciamento de senhas: permite o gerenciamento de senhas criptografadas.

Cabe ressaltar que este capítulo teórico apresenta os tópicos mais importantes da proposta ODP, dedicou-se especial atenção àqueles tópicos referentes à arquitetura de especificação de sistemas, cujo conteúdo é o subsídio principal dentro do escopo desta tese.

Este trabalho não abrange a parte formal da proposta ODP, mas pode-se obter mais informações detalhadas nas seguintes referências bibliográficas: [Gotzhein, 95], [Najm, 95].

A seguir no capítulo seguinte, os conceitos e a especificação ODP são colocados de forma sistemática, dentro de um processo metodológico com o objetivo de ser utilizado nos projetos de sistemas de automação.

Capítulo 3 Metodologia de Projeto ODP

Neste capítulo apresenta-se a metodologia que será utilizada no desenvolvimento do sistema aberto de automação aplicado a projetos dos grandes empreendimentos.

O conteúdo deste capítulo abrange a definição do modelo de referência de sistema aberto de automação (SAA) e, principalmente, apresenta a estrutura da metodologia de projeto ODP.

Cabe ressaltar que esta metodologia possui uma estrutura conceitual baseada no padrão ODP e utiliza os conceitos e modelos da metodologia orientada a objeto.

3.1 Modelo de Referência SAA

O modelo de sistema aberto de automação, apresentado em [Martucci, 92], foi o início de um conjunto de pesquisas correlatas dentro da área de automação, sua utilização representou uma atividade comum em todos os trabalhos.

Mas, cabe ressaltar que existem outros modelos de sistemas de automação como os encontrados na referências [Becerra, 93] e que também poderão ser utilizados pois possuem os mesmos objetivos e terminam por estruturar a corporação como um todo.

O modelo SAA é denominado, neste trabalho de tese, de modelo de referência SAA com o intuito de ressaltar sua importância gerada no âmbito da pesquisa na área de automação, especialmente no grupo de pesquisa do autor, como foi explicado anteriormente no item 1.7.

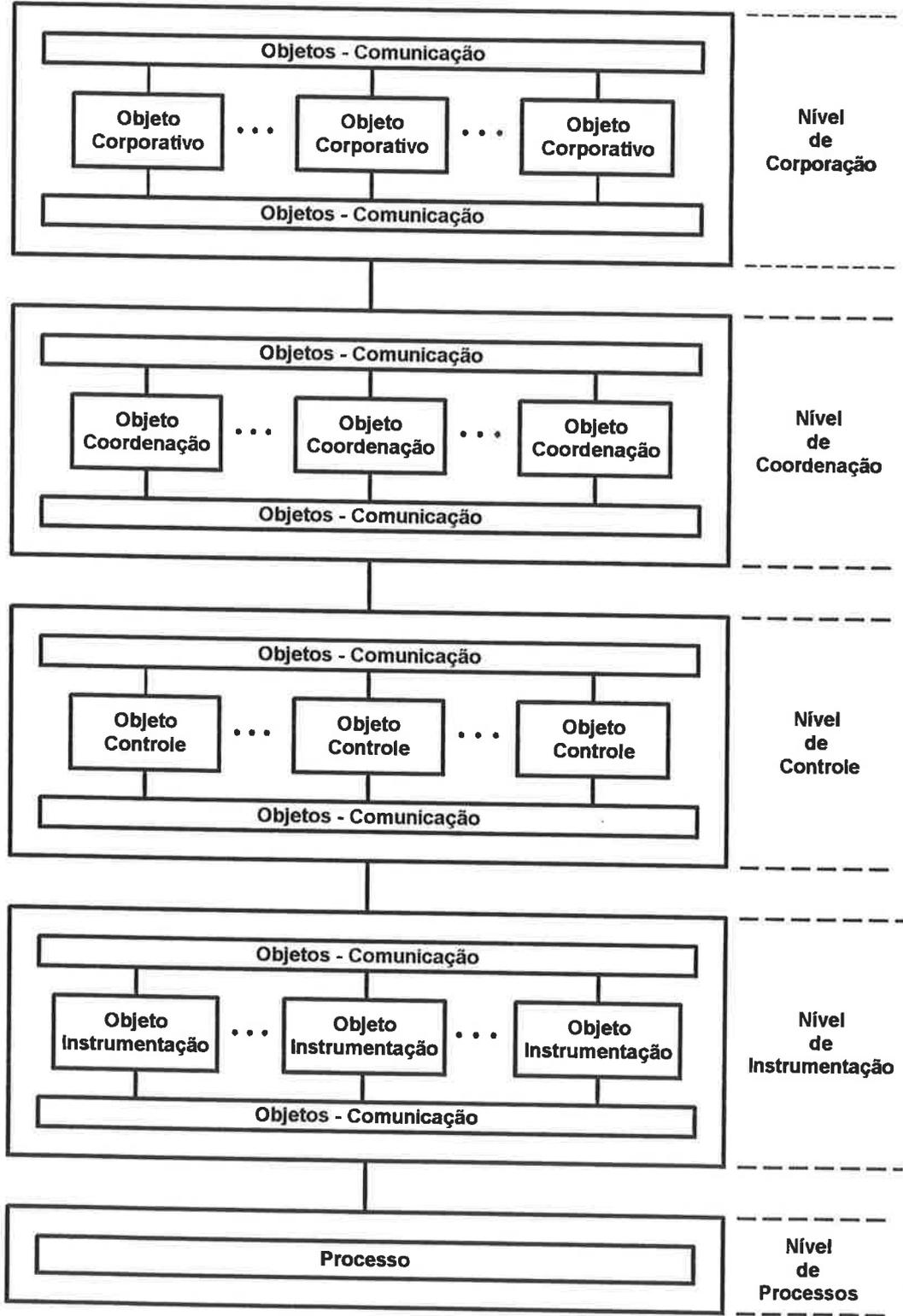


Figura 3.1 Modelo de Referência SAA

Também recebe este nome, porque representa a estrutura inicial da metodologia ODP aqui proposta, deixa-se claro que este modelo não representa um padrão ou norma dentro da área da automação.

O modelo de referência de sistema aberto de automação (SAA), mostrado na figura 3.1, é um modelo abstrato de classes de objetos, originada da proposta apresentada na referência bibliográfica [Martucci, 92] e que foi utilizada nos trabalhos, [Becerra, 93], [Tanomaru, 94] e [Matsuyama, 97].

Este modelo de referência, visa a definição de arquiteturas abertas que poderão ser implementados em diversos domínios de aplicação da automação. Exemplos desta diversidade, encontram-se no trabalho de [Matsuyama, 97], onde foi determinada arquiteturas de automação aplicados a processos contínuos, processos discretos, automação bancária e automação predial, e no trabalho de [Becerra, 93], onde foi aplicado na automação da manufatura.

Estas arquiteturas abertas de automação possuem altos graus de interoperabilidade, interconectividade e portabilidade e permitem suportar elementos para configurar um ambiente cooperativo [Becerra, 96b].

3.1.1 Características



O modelo de referência SAA possui quatro características importantes: é uma estrutura abstrata, é uma estrutura hierárquica, é um sistema aberto e um sistema inerentemente distribuído.

Este modelo representa uma estrutura abstrata, isto é, é uma representação em alto nível do sistema de automação, apresentando só as características relevantes à definição do sistema.

Esta característica facilita a tarefa do projetista de sistemas abertos de automação, pois ele consegue mapear no modelo, os diversos aspectos que

devem ser manipulados no desenvolvimento de um sistema distribuído e aberto como gerenciamento, comunicação, processamento e tecnologia, [Garrahan, 93].

O modelo está composto de níveis hierárquicos, esta divisão permite a localização e distribuição adequada das complexidades envolvidas nos requisitos do sistema, tais como: funções, objetos, capacidade de processamento, controle, comunicação, confiabilidade e disponibilidade, interface homem-máquina, gerenciamento de dados, entre outros requisitos importantes.

O modelo é um sistema aberto porque seus elementos e interfaces são definidos por padrões internacionais e padrões de fato, sendo que quando a aplicação não possui o grau de abertura adequada, quer dizer a disponibilidade de alguma solução padronizada, pode-se incluir especificações proprietárias [Becerra, 93].

O modelo é um sistema distribuído, significa que capacidades de processamento, armazenamento e comunicação dos diferentes elementos que o constituem, estão distribuídos computacionalmente ou geograficamente dentro do sistema de automação.

3.1.2 Estrutura do SAA

O modelo de referência SAA está composto dos seguintes elementos: as classes de objetos denominados de objetos automação e os níveis hierárquicos, que por sua vez estão compostos de camadas funcionais.

3.1.2.1 Classes de Objetos Automação

As classes de objetos aqui denominadas, por simplicidade, de objetos automação são entidades conceituais que realizam suas atividades nos

diferentes níveis hierárquicos do modelo. Estas classes encapsulam seus estados e interagem através de interfaces corretamente definidas, [Rumbaugh, 92], [Booch, 94].

A figura 3.2.a mostra a estrutura hierárquica dos objetos automação, o topo da figura representa a classe principal e os objetos dos níveis inferiores representam os tipos de objetos automação que existem nos sistema.

A figura 3.2.b apresenta, como exemplo, instâncias dos objetos automação de dois níveis hierárquicos, de instrumentação e de coordenação, também pode-se ver nesta figura que um objeto possui três elementos básicos:

- Identificador: representa o nome ou a identificação do objeto automação no domínio do sistema de automação;
- Atributo: definem os requisitos específicos de funcionamento do objeto automação, como por exemplo, tempos de resposta de um objeto sensor e os índices de confiabilidade;
- Serviço: são operações realizadas que são oferecidas aos outros objetos automação e cujo resultado originam mudanças nos estados dos objetos.

3.1.2.2 Níveis Hierárquicos

O SAA da figura 3.1, define os seguintes níveis hierárquicos: o nível de processo, o nível de instrumentação, o nível de controle, o nível de coordenação e o nível corporativo. A seguir apresenta-se estes níveis:

- Nível corporativo: está constituído de objetos automação denominados de objetos corporação que participam das atividades envolvidas em processos de alto nível empresarial, como a interação entre subsidiárias corporativas.

Estes elementos oferecem serviços que correspondem às atividades de gerenciamento das estratégias e políticas a nível global e as atividades de gerenciamento da informação corporativa;

- Nível de coordenação: está constituído de objetos automação denominados de objetos coordenação, que oferecem serviços que correspondem ao gerenciamento das informações armazenadas nas bases de dados centrais, gerenciamento de estratégias e políticas da produção, interfaceamento homem-máquina a nível central, gerenciamento das configurações;
- Nível de controle: está constituído de objetos automação denominados objetos controle, que oferecem serviços que correspondem as atividades de gerenciamentos dos objetos do nível de instrumentação, gerenciamento das atividades no âmbito do chão de fábrica ou do processo, interfaceamento homem-máquina a nível local, execução de algoritmos de controle e de intertravamentos;
- Nível de instrumentação: está constituído de objetos automação denominados de objetos instrumentação, oferecem serviços que correspondem as atividades de comunicação com objetos processo do nível de processo, gerenciamento dos instrumentos, interfaceamento homem-máquina no nível do processo, execução de algoritmos de controle;
- Nível de processo: engloba os objetos processos que constituem os principais processos que a empresa suporta, este nível não faz parte do SAA:

Cabe ressaltar que este modelo representa a divisão lógica do sistema de automação, do ponto de vista da implementação alguns níveis hierárquicos podem ser integrados em um mesmo equipamento ou sistema.

As características específicas dos objetos automação dos diferentes níveis hierárquicos para diferentes aplicações poderão ser encontrados nas referências bibliográficas [Martucci, 92], [Becerra, 93], [Tanomaru, 94], [Silva, 96] e [Matsuyama, 97].

3.1.2.3 Camadas Funcionais

Cada nível hierárquico está dividido por zonas denominadas camadas funcionais. Estas camadas tem por objetivo, organizar e localizar adequadamente os objetos automação dentro de cada nível hierárquico, e se considera dois aspectos: as funções específicas do nível hierárquico e as funções de comunicação.

A figura 3.3, mostra a camada de processamento e as camadas de comunicação do nível de controle de um sistema de automação.

A camada denominada de camada do processamento engloba os objetos automação que executam exclusivamente funções características de cada aplicação, por exemplo um objeto coordenação que oferece o serviço de gestão de políticas externas.

A camada de comunicação engloba os objetos automação que suportam exclusivamente as funções de comunicação entre níveis hierárquicos, por exemplo um objeto instrumentação que oferece os serviços do protocolo via de campo (FieldBus) [Muñoz, 98].

A estrutura do modelo de referência SAA permite a integração do sistema de automação como um todo, suportando o sistema de informação da corporação e visando o fluxo contínuo da informação, aliás, sua forma abstrata torna este modelo o ponto inicial da metodologia proposta a seguir.

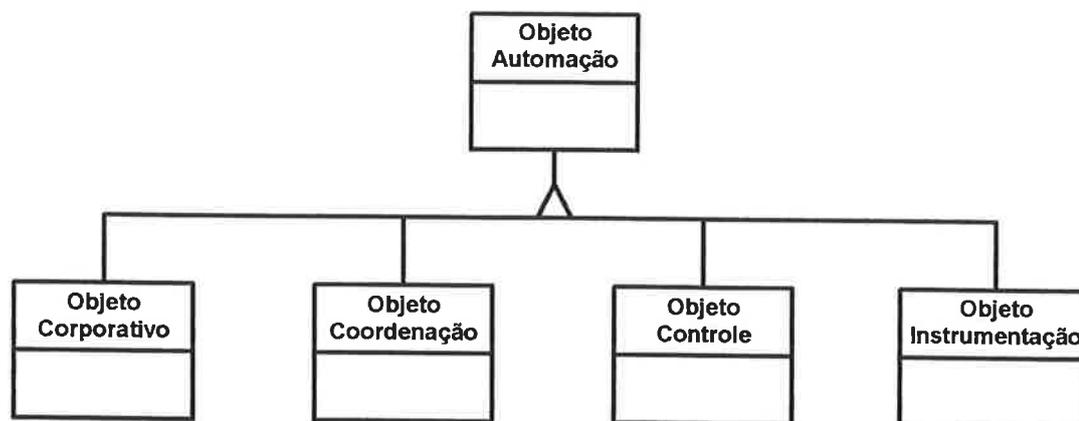


Figura 3.2 a Estrutura Hierárquica dos Objetos Automação

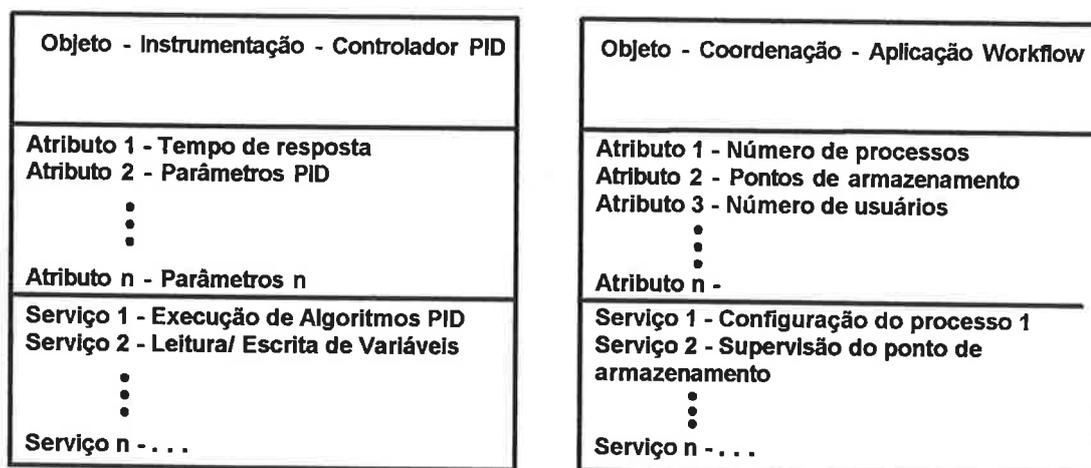


Figura 3.2 b Instâncias dos Objetos Instrumentação e Coordenação

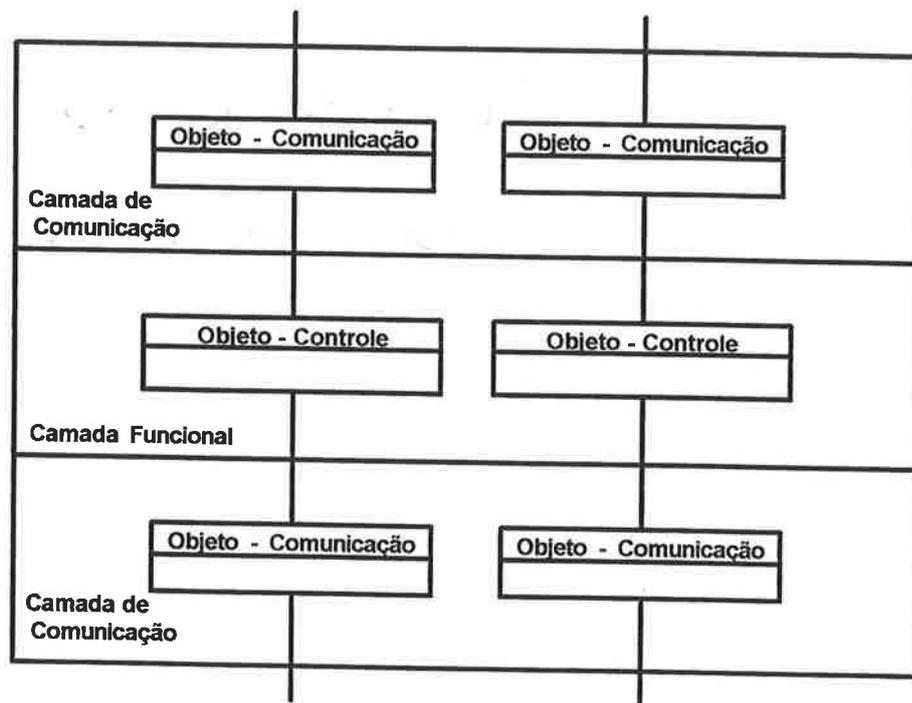


Figura 3.3 Camadas Funcionais do Nível Hierárquico de Controle

3.2 Estrutura Metodológica ODP

A metodologia de desenvolvimento ODP é um conjunto de conceitos, procedimentos e modelos, cuja base conceitual está definida no padrão ODP apresentado no capítulo dois deste trabalho de tese.

Também são utilizados a estrutura conceitual das metodologias orientadas a objetos (MOO) e o embasamento teórico proposto na arquitetura CORBA, [UML, 97], [Rumbaugh, 91], [Booch, 94], [Eckert, 96], [Lan, 92], [Jacobson, 92].

Dada a complexidade inerente dos projetos de sistemas distribuídos e abertos, esta proposta tem por finalidade principal, criar um esquema de projeto organizado, seqüencial e sistemático de forma a facilitar as tarefas de gerenciamento e desenvolvimento e obter altos graus de produtividade, qualidade e custos adequados no projeto.

Nos próximos itens, será definida a metodologia de forma geral e será apresentada um detalhamento de cada fase da metodologia.

3.2.1 Definição e Características

Existem diversas formas de obter uma arquitetura ou implementar uma automação, entretanto, todas estas formas levam ao mesmo objetivo, mas com diferentes resultados.

Para poder atingir corretamente as metas de um projeto, o projetista utiliza conceitos, procedimentos, critérios práticos ou heurísticos, e ferramentas; acima de tudo, preocupa-se em observar o desenvolvimento do projeto como um todo e não suas partes isoladas.

Dentro deste contexto, a metodologia de projeto ODP é um conjunto de conceitos, procedimentos, ferramentas e critérios organizados sistematicamente, onde o processo de desenvolvimento é observado como um todo, permitindo o controle da qualidade e a consistência do projeto, [Sinderen 95].

Esta metodologia ODP possui as seguintes características: engloba conceitos e procedimentos ODP e MOO, define abstrações, tem um ciclo de vida, e se adapta a diversos domínios de aplicação dentro da automação.

Os conceitos da metodologia ODP estão divididos em categorias, que são as seguintes:

- Conceitos referentes à modelagem: onde são definidos os conceitos de objetos, encapsulamento, abstração, comportamento, estado e interfaces;
- Conceitos na especificação: onde são definidos os conceitos de composição, moldes de especificação (templates), funções ou papéis desempenhados, tipos e hierarquia de classes;
- Conceitos na arquitetura: onde são definidos os conceitos de identificação, grupos, domínios, contratos e conexão.

No que diz respeito a abstração, a metodologia ODP define a abstração como um mecanismo que permite selecionar as características mais relevantes do sistema, e que são necessárias no instante da definição do sistema [Jordan, 91].

Na metodologia ODP as abstrações são denominadas de pontos de vista, e existem cinco pontos de vista: o ponto de vista de empresa, o ponto de vista da informação, o ponto de vista da computação, o ponto de vista da engenharia e o ponto de vista tecnológico [Bond, 97].

Em relação ao ciclo de vida, esta metodologia utiliza, de forma geral, o modelo de ciclo de vida denominado de cascata que é uma estrutura seqüencial e sistemática, em cada fase utiliza-se o ciclo de vida em espiral para realizar contínuas atualizações [Rakos, 90].

Cabe ressaltar que existem outros modelos de ciclo de vida que poderão ser adaptados na metodologia mas foi selecionado o modelo de cascata e em espiral por sua simplicidade e devido aos resultados obtidos em [Tanomaru, 94] e [Matsuyama, 97].

Em termos da adaptação, esta metodologia visa ser utilizada em diferentes domínios de aplicação, existem dois aspectos que contribuem para esta característica: a definição dos pontos de vista que podem ser definidos em quaisquer aplicações, não existe restrição da área; e a utilização do modelo de referência SAA que por definição pode ser aplicado em diferentes áreas também sem restrição [Leydekkers, 96].

Com as características supracitadas, esta metodologia ODP assegura a capacidade de suportar os requisitos e objetivos das diversas áreas de aplicação.

É bom indicar, que uma metodologia de desenvolvimento deve definir uma semântica, isto é, deve definir uma linguagem de especificação formal que represente os conceitos e requisitos de especificação de forma precisa e completa [Sinderen, 95].

Neste sentido esta tese não implementa esta característica, apesar que o padrão ODP recomenda a definição de uma arquitetura da semântica baseada em linguagens de especificação formal [ISO, 95d].

3.2.2 Fases de Desenvolvimento

A figura 3.4 mostra a estrutura da metodologia ODP para desenvolver sistemas distribuídos e abertos de automação.

Nesta figura, as elipses representam as fases do desenvolvimento, onde são realizadas atividades específicas de projeto; as flechas representam o fluxo das informações do projeto e a passagem de subprodutos de cada fase; os pontos em cada flecha representam os pontos de controle .

A atividade de desenvolvimento inicia-se com a estruturação do modelo de referência SAA, segue a continuação a especificação ODP, com a definição da arquitetura, neste ponto a seqüência é desdobrada em vários caminhos, e inicia-se o projeto ODP de cada nível hierárquico, terminando na implementação e operação do sistema de automação.

As fases de desenvolvimento da metodologia ODP são as seguintes:

- Definição do modelo de referência SAA: deve-se realizar atividades de análise de sistema como são a detecção das funções dos níveis hierárquico do sistema de automação, a determinação da estrutura organizacional, a determinação da infra-estrutura computacional, e a definição das informações manipuladas no sistema;

O objetivo desta fase é obter um modelo abstrato do sistema aberto de automação, para poder obter informações que possam facilitar a fase de especificação ODP;

- Especificação ODP: deve-se realizar basicamente a definição dos cinco pontos de vista: empresa, informação, computacional, engenharia e tecnológico, deve-se gerar uma série de modelos de objetos representativos de cada ponto de vista e uma especificação em linguagem natural;

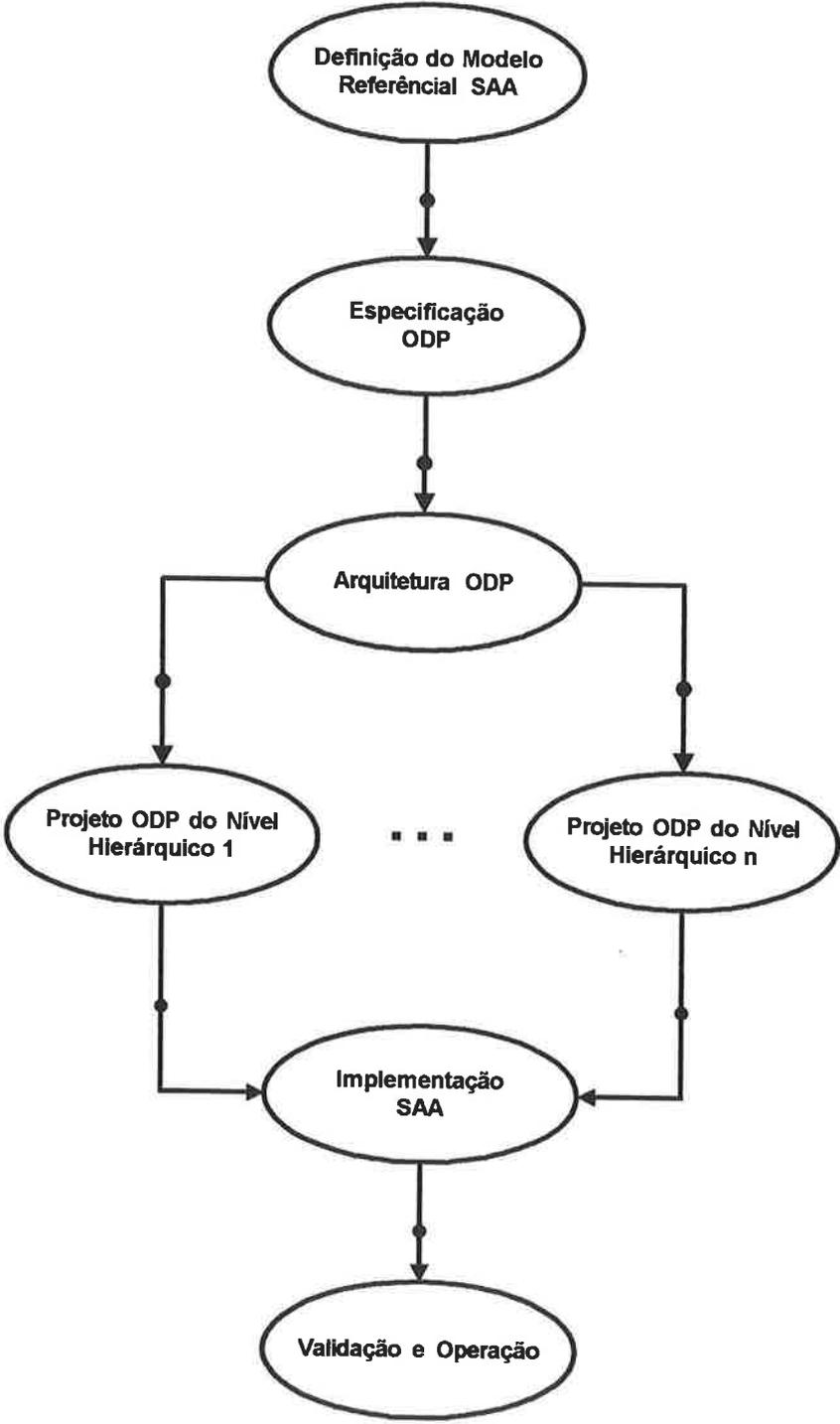


Figura 3.4 Estrutura da metodologia ODP

Nível de Abstração	Fases da Metodologia	Ferramentas	Produtos
Domínio da aplicação	Definição do SAA	Modelo SAA	Especificação do SAA Diagrama SAA
Requisitos da empresa	Especificação do ponto de vista da empresa	<i>Uses cases</i> Diagrama de classes	Regras da empresa Modelo OO
Requisitos da informação	Especificação do ponto de vista da informação	Contratos Diagrama de classes Diagrama de colaboração Diagrama de estado	Regras da informação Modelo OO
Requisitos da computação	Especificação do ponto de vista da computação	Contratos <i>Uses cases</i> Diagrama de classes	Regras da computação Modelo OO
Requisitos da engenharia	Especificação do ponto de vista da engenharia	<i>Uses cases</i> Contratos Diagrama de classes Diagrama de estado	Regras da engenharia Modelo OO
Requisitos da tecnologia	Especificação do ponto de vista da tecnologia	Padrão de Especificação Diagrama de classes	Documento da tecnologia Modelo OO
Subsistemas	Arquitetura ODP	Modelo de Arquitetura ODP	Diagrama de arquitetura dos níveis hierárquicos
Processadores	Projeto ODP	Análise OO Projeto OO	Modelos OO Modelo de processadores
Técnico	Implantação e validação	Estratégias corporativas	SAA corporativo

Tabela 3.1 Resumo da Estrutura da Metodologia ODP

- Definição da arquitetura ODP: deve-se determinar os objetos da estrutura arquitetônica e definir os mecanismos de abstração implementados pelas funções ODP ;
- Projeto ODP: são definidos de forma específica os objetos automação de cada nível hierárquico, utilizando as técnicas da análise e projeto orientado a objeto;
- Implementação e validação: engloba as tarefas de implementação de cada nível hierárquico e envolve as atividades de integração e validação do sistema como um todo.

A tabela 3.1 apresenta um resumo de todas as fases da metodologia ODP incluindo ferramentas e produtos finais.

3.2.3 Formas de Aplicação

A metodologia ODP possui graus de flexibilidade nas formas de aplicação, isto é, suportar modificações para se adaptar a cada aplicação. O modelo de referência SAA é a chave que determina as formas de aplicação da metodologia ODP.

Desta forma, as várias formas de aplicação estão nos seguintes pontos: aplica-se a estrutura metodológica ao sistema de automação como um todo incluindo todos os níveis hierárquicos, aplica-se a um determinado nível hierárquico, aplica-se a uma camada funcional específica, aplica-se a um sistema aplicativo ou aplica-se a um objeto automação específico do sistema.

3.3 Especificação ODP

A seguir será apresentada a fase da metodologia denominada de especificação ODP onde são definidos os requisitos do sistema de automação segundo os cinco pontos de vista.

Nesta fase serão recomendados os procedimentos para definir os modelos de objetos e serão definidas as regras de especificação dos cinco pontos de vista.

Os modelos de objetos são representações baseadas na proposta [UML, 97] e [Rumbaugh, 91], e as regras de especificação ou de estruturação são estruturas descritivas em linguagem natural (textos) .

Cabe indicar que o modelo de referência SAA deverá gerar informações necessárias para a identificação dos requisitos, dos processos e dos objetos e outros elementos da especificação.

3.3.1 Especificação do Ponto de Vista Empresa

A especificação do ponto de vista empresa define o sistema da visão de empresa, e deve representar a definição dos requisitos que a empresa impõe ao sistema que está sendo desenvolvido.

Esta especificação está constituída de duas atividades: a definição das regras de empresa e o modelo de objetos empresa.

Estas atividades deverão ser realizadas segundo o modelo de ciclo de vida denominado de espiral, realizando uma retroalimentação controlada da modelagem para a definição de regras de empresa.

3.3.1.1 Definição das Regras de Empresa

As regras de empresa ou de estruturação são estruturas descritivas em linguagem natural que definem os processos, o papel que desempenha cada usuário, as políticas externas e internas, entre outros elementos, mas dentro do contexto empresarial.

No ponto de vista da empresa devem ser definidas as seguintes regras de empresa:

- Regras dos usuários: definem os identificadores, as obrigações, as proibições, e as permissões que serão configurados na estrutura dos objetos empresa;
- Regras de interação: definem os processos envolvidos na interação entre objetos empresa, estas regras contém políticas que limitam as atividades de negociação, de intercomunicação, de intracomunicação e de segurança entre objetos empresa;
- Regras dos recursos: definem as políticas de gerenciamento dos recursos do sistema, estes recursos são elementos distribuídos que oferecem diferentes serviços aos objetos empresa, desta forma contribuem na execução de tarefas cooperativas. Neste sentido devem-se definir as regras de configuração de recursos, regras de utilização e regras de desativação dos recursos;
- Regras de relações externas e internas: definem-se as políticas que limitam a relação entre empresa e o meio ambiente externo, e as políticas internas referentes à comunidade, o domínio e a federação que limitam o comportamento dos objetos empresa. Entre as principais regras que deverão ser especificadas estão as regras referentes à segurança, as regras que englobam as normas legais e as regras de utilização de padrões ou normas específicas;

- Regras de negócios da empresa: definem-se as políticas que são utilizadas na empresa para atingir metas referentes aos negócios que compõem o escopo principal da empresa.

A especificação das regras de empresa utiliza como ferramenta de apoio os denominados *Uses Cases*, os quais são apresentados em [UML, 97], [Larman, 97].

Os *Uses Cases* são documentos em linguagem natural que descrevem as seqüências de eventos entre atores ou agentes ou objetos empresa quando interagem no sistema. Os diagrama são representações gráficas mostrando os elementos básicos destes textos facilitando seu entendimento.

A estrutura básica de um *Uses Cases* está constituída dos seguintes elementos: os atores, um nome específico, o tipo ou classificação específica e uma descrição clara da interação entre atores e sistema.

Esta estrutura não é rígida e pode incluir novos elementos mas depende do espírito da documentação em relação à clareza da comunicação.

3.3.1.2 Modelagem do Ponto de Vista da Empresa

A modelagem do ponto de vista da empresa visa a obtenção de um diagrama de objetos empresa onde é mostrado a interação destes elementos segundo o contexto definido pelas regras de empresa.

Como ferramenta de modelagem utilizam-se diversos modelos definidos na referência [UML, 97].

O modelo básico desta atividade é denominado diagrama de estrutura estática que é um diagrama de objetos empresa onde são mostrados as associações e os atributos destes elementos.

Dependendo da exigência da modelagem ou quando o escopo do sistema é voltado especificamente para a visão da empresa, recomenda-se adicionar o diagrama de seqüências de eventos, que é um tipo de diagrama mostrando a evolução dos eventos que ocorrem no tempo [UML, 97].

Para o processo de identificação dos objetos de empresa utilizam-se as informações definidas nas regras de empresa, e aplica-se quaisquer métodos de identificação propostos em [Booch, 94].

3.3.2 Especificação do Ponto de Vista Informação

A especificação do ponto de vista da informação apresenta os requisitos do sistema segundo a informação manipulada e armazenada pelos objetos empresa e pelo sistema ODP.

A especificação do ponto de vista da informação engloba duas etapas: a definição das regras de informação e a modelagem de objetos informação.

3.3.2.1 Definição das Regras de Informação

Esta etapa define três tipos de regras de informação segundo os esquemas de informação definidos no capítulo dois. A seguir são definidos estas regras:

- Regras invariantes: determinação das regras que especificam o comportamento dos esquemas invariantes de informação;

- Regras estáticas: determinação das regras que definem os esquemas estáticos de informação, isto é, definem os estados característicos dos objetos informação em um determinado tempo;
- Regras dinâmicas: determinação das regras que definem os esquemas dinâmicos de informação, isto é, definem todos os estados dos objetos informação.

As regras de informação utilizam como ferramenta de apoio para especificação os denominados contratos. Os contratos são documentos de especificação que descrevem a forma de como a operação chega a um resultado final [Larman, 97].

Os contratos de informação descrevem as mudanças de estados de um objeto informação quando uma operação é realizada no sistema ODP.

Estes contratos possuem os seguintes elementos: nome, objetivo, tipo de esquema, saída, pre-condição e a pós-condição.

3.3.2.2 Modelagem do Ponto de Vista da Informação

Esta atividade encarrega-se da modelagem da informação do sistema, para este fim utiliza-se os diagramas do ponto de vista da empresa e as regras de informação do sub-item anterior.

Dado que existem três tipos de esquemas de informações definidos pelo padrão ODP, serão utilizados dois tipos de diagramas apresentados em [UML, 97].

Para modelar os esquemas invariantes utiliza-se um diagrama de estruturas estáticas que mostrará os objetos informação com suas respectivas associações.

Na modelagem dos esquemas estáticos e dinâmicos serão utilizados os diagramas de colaboração que apresentam os objetos informação ilustrando as informações transmitidas e recebidas [Larman, 97].

Dado que estes dois esquemas possuem uma relação direta com o tempo, em determinados casos poderão ser utilizados os diagramas de estados apresentados em [Rumbaugh, 91] e [UML, 97].

3.3.3 Especificação do Ponto de Vista Computação

A especificação do ponto de vista computação descreve como as aplicações e os componentes distribuídos do sistema ODP interagem de forma cooperativa e transparente.

A especificação do ponto de vista computação deve gerar um documento de especificação denominado de regras da computação e um modelo de objetos cujos elementos são denominados de objetos computação.

Estes dois produtos devem mostrar informações referentes à configuração dos objetos computação, seus comportamentos internos, e as interações entre objetos computação, não deve mostrar os objetos que suportam a comunicação.

Esta especificação está composta das seguintes atividades: a definição das regras da computação e a modelagem do ponto de vista da computação.

3.3.3.1 Definição das Regras da Computação

As regras da computação representam a transformação das regras de estruturação, apresentadas no padrão ODP [ISO, 95c], em documentos de especificação.

Dentro deste escopo, deve-se definir as seguintes regras da computação: regras de interação, regras de conexão, regras de estruturação e regras de falhas.

3.3.3.1.1 Regras de Interação

São as regras que especificam a interação entre as interfaces dos objetos computação, estas regras estão relacionadas com os três tipos de interfaces definidas no capítulo dois: interfaces para sinais, interfaces para fluxos, interfaces para operações. Estas regras são as seguintes:

- Regras dos sinais: definem a forma de iniciação e da resposta ao sinal entre o iniciador e o respondedor;
- Regras dos fluxos: definem o esquema de geração e recebimento de fluxo de informação entre o produtor e consumidor;
- Regras das operações: definem o esquema de geração e recebimento das mensagens ou operações entre o cliente e o servidor.

3.3.3.1.2 Regras de Conexão

As regras de conexão são referentes ao tipo de conexão configurado entre dois objetos computação, e utilizam como ponte, entre os dois, um terceiro objeto denominado de objeto conexão. Devem ser definidas as seguintes regras de conexão:

- Regra de conexão primitiva: especifica o esquema de conexão entre dois objetos computação que implementam interfaces iguais;

- Regra de conexão composta: especifica o esquema de conexão entre vários objetos computação e um objeto conexão.

3.3.3.1.3 Regras de Estruturação

As regras de estruturação especificam a estrutura funcional e comportamental dos objetos computação e dos objetos conexão.

A especificação destas regras são identificadas com os serviços oferecidos pelos objetos computação, mas englobam outras características como atributos e estados.

Entre os principais serviços que devem ser especificados nas regras de estruturação estão:

- Serviços que permitem a manipulação de sinais, operação e fluxo de informação;
- Serviços que permitem a configuração e a desativação de interfaces;
- Serviços que permitem a manipulação dos estados do objeto computação;
- Serviços que se encarregam de processamentos específicos.

Na configuração específica do objeto conexão encontram-se os seguintes serviços:

- Serviço que implementa os mecanismos para supervisionar a utilização e as mudanças da conexão;
- Serviço que implementa mecanismos que autorizam as mudanças da conexão;

- Serviço que manipula as mudanças no que diz respeito à qualidade do serviço;
- Serviço que se encarrega da desativação da conexão como um todo;

3.3.3.1.4 Regras de Falhas

As regras de falhas especificam os mecanismos de falhas de determinados eventos que são determinados a partir da análise da especificação dos serviços dos objetos computação e dos objetos conexão.

Deve-se estruturar as seguintes regras de falhas básicas:

- Regras de comunicação: especificam mecanismos que manipulam as falhas que ocorrem na interação entre objetos computação;
- Regras de segurança: mecanismo que se responsabilizam pelas falhas que ocorrem com os diversos aspectos da segurança do sistema;
- Regras dos recursos; gerenciam as falhas que ocorrem nos objetos recursos do sistema.

As regras da computação utilizam como ferramenta de apoio os documentos denominados contratos, utilizado no ponto de vista da informação, e no caso especial das regras de estruturação utiliza-se os *Uses Cases*

3.3.3.2 Modelagem do Ponto de Vista Computacional

A modelagem do ponto de vista computacional visa a produção de modelos de objetos que apresentam os objetos computação, os objetos conexão e suas respectivas associações.

Para representar este ponto de vista utiliza-se um diagrama de estruturas estáticas ou diagramas de classes e quando for necessário apresentar o comportamento do sistema recomenda-se elaborar um diagrama de colaboração.

3.3.4 Especificação do Ponto de Vista da Engenharia

A especificação do ponto de vista da engenharia define os mecanismos e as funções necessárias para suportar a distribuição dos objetos computação do sistema ODP.

Este ponto de vista gera um documento de especificação denominado de regras de engenharia e origina um conjunto de diagramas de objetos constituído de objetos engenharia e suas respectivas associações.

Cabe ressaltar que esta especificação utiliza parte da especificação do ponto de vista computação porque existe uma correspondência direta com este ponto de vista. Lembrando que os objetos computacionais transformam-se em objetos básicos de engenharia.

O procedimento de elaboração desta especificação possui duas etapas: a definição das regras de engenharia e a modelagem de objetos engenharia.

3.3.4.1 Regras de Engenharia

As regras de engenharia que devem ser determinadas são: regras dos canais, regras das interfaces referenciais, regras de conexão distribuída, regras de alocação, regras da estrutura de gerenciamento e regras de falhas.

3.3.4.1.1 Regras dos Canais

Os canais suportam de forma transparente a interação distribuídas entre objetos engenharia; e são especificados pela definição das seguintes regras:

- Regras da estrutura: definem a estruturação do canal constituído de objetos engenharia denominados de adaptador, conector, protocolo.

A estrutura do canal, entre dois objetos básicos de engenharia, deverá seguir as seguintes seqüências padrão: adaptador - conector - protocolo – protocolo – conector - adaptador; e adaptador - conector – protocolo – interceptor – protocolo – conector - adaptador;

- Regras do adaptador: especificam os serviços que realizam a conversão dos dados, e especifica as interfaces com objetos engenharia básicos, com o objeto conector e com as interfaces de controle;
- Regras do conector: especificam o serviço de realocização, o serviço de controle da comunicação, e as interfaces com objetos engenharia, com o objeto adaptador, com o objeto protocolo e com as interfaces de controle;
- Regras do protocolo: especificam o serviço de comunicação, a interface com objetos engenharia, a interface com o objeto conector, e com o objeto interceptor, e as interfaces de controle;
- Regras do interceptor: especificam o serviço de transformação de protocolos, as interfaces com os objetos protocolos e a interface de controle.

As interfaces de controle dos objetos engenharia do canal são elementos que permitem o gerenciamento dos elementos do canal.

3.3.4.1.2 Regras das Interfaces Referenciais

As interfaces referenciais são especificações padronizadas das interfaces dos objetos engenharia, esta padronização é realizada com o objetivo de facilitar o gerenciamento das interfaces no espaço e no tempo dentro do sistema distribuído e aberto.

As regras das interfaces referenciais devem especificar os seguintes elementos: o tipo de interface, o identificador das interfaces, o endereço das interfaces, e os dados para viabilizar os mecanismos de detecção e reparo da conexão.

3.3.4.1.3 Regras de Conexão Distribuída

Esta regra especifica a seqüência de eventos necessária para o estabelecimento do canal. A atividade de especificar envolve os objetos engenharia e o objeto núcleo do nó.

Estas regras devem especificar os serviços de inicialização e o serviço de criação dos objetos que compõem o canal.

3.3.4.1.4 Regras de Relocalização

Estas regras especificam a seqüência de eventos que ocorrem quando é executado o serviço de relocalização.

Estas regras definem todos os tipos de relocalização, isto é, existem regras para reativação e desativação de objetos, quando ocorre a uma notificação de eventos e recuperação, quando ocorre uma migração e quando são executadas funções de gerenciamento da comunicação que mudam o identificador da interface.

3.3.4.1.5 Regras da Estrutura de Gerenciamento

As regras da estrutura de gerenciamento permitem a estruturação dos domínios gerenciais do sistema, especificando os seguintes elementos: o grupo, a cápsula e o nó. As regras da estrutura são as seguintes:

- Regras do grupo: definem as políticas de interação dentro do grupo, assim como, especificam a interação dos objetos básicos de engenharia com o objeto gerenciador do grupo e com o objeto gerenciador do nó;
- Regras da cápsula: definem as políticas de interação dos objetos de engenharia dentro da cápsula e especifica a interação entre os objetos gerenciadores de grupos e o objeto gerenciador da cápsula;
- Regras do nó: definem as políticas de interação dentro do nó, especificando a interação entre o objeto gerenciador do nó ou o objeto núcleo e os outros objetos gerenciadores dos outros níveis da estrutura gerencial.

3.3.4.1.6 Regras de Falhas

As regras de falhas do ponto de vista da engenharia contemplam a especificação das seguintes falhas:

- Regras que especificam o serviço de manipulação das falhas do grupo, este serviço deve estar configurado na estrutura do objeto gerenciador do grupo;
- Regras que especificam o serviço de manipulação de falhas da cápsula, este serviço deve ser configurado no objeto gerenciador da cápsula;
- Regras que especificam o serviço de manipulação das falhas do nó, este serviço deve ser configurado no objeto protocolo do outro nó interconectado;

- Regras que especificam as falhas de comunicação no domínio da comunicação, devem ser manipuladas pelos objetos protocolo do outro domínio interconectado.

Estas regras de engenharia utilizam como ferramentas de apoio os *Uses Cases* e os documentos denominados contratos, apresentados em [UML, 97] e [Larman, 97].

As regras dos canais, interfaces referenciais, realocização e falhas utilizam os contratos e as regras de conexão distribuída e as regras das estruturas de gerenciamento utilizam os *Uses Cases*.

3.3.4.2 Modelagem do Ponto de Vista da Engenharia

A modelagem do ponto de vista da engenharia apresenta um modelo de objetos constituído de objetos básicos de engenharia e objetos de engenharia que constituem o canal de comunicação.

Este ponto de vista deve utilizar um diagrama de estruturas estáticas ou diagramas de classes para representar os requisitos de engenharia.

O canal de comunicação poderá utilizar para completar sua definição um diagrama de estados apresentado em [Rumbaugh, 91] e [UML, 97].

3.3.5 Especificação do Ponto de Vista Tecnológico

A especificação do ponto de vista tecnológico contém os requisitos referentes à tecnologia, que serão utilizados na implementação dos elementos do ponto de vista da engenharia. Nesta especificação deverão ser realizadas as seguintes atividades:

- Seleção da tecnologia que implementa os objetos engenharia, dando prioridade às especificações de sistemas ODP, e optando por soluções proprietárias quando for necessário [Becerra 93];
- Identificação das restrições que apresentam os outros pontos de vista, isto é, define-se regras de restrição originadas nos modelos de objetos dos outros pontos de vistas;
- Construção de um modelo de objetos do ponto de vista tecnológico, este modelo será um diagrama de classes ou diagrama de estrutura estática, cujos elementos são os objetos tecnologia.

Neste item foi apresentado a fase da metodologia proposta denominada a especificação ODP, no próximo item deve-se definir a fase da arquitetura como uma consequência desta especificação.

3.4 Arquitetura ODP

O modelo de referência SAA e a especificação ODP, baseada nos cinco pontos de vista, impõem requisitos e restrições importantes para definir a arquitetura de um sistema distribuído e aberto aplicado a automação.

Neste item serão apresentadas as características da arquitetura ODP e a especificação das funcionalidades ODP.

Esta arquitetura deverá ser configurada nos diferentes níveis hierárquicos do sistema aberto de automação.

3.4.1 Considerações Iniciais

Antes de apresentar a estrutura da arquitetura, serão apresentados dois pontos importantes: a importância da arquitetura e a relação da arquitetura com o modelo SAA e com a especificação ODP.

Uma metodologia de desenvolvimento de projetos possui uma estrutura composta de uma série de atividades agrupadas em fases, esta estrutura organizada permite a evolução gradual do sistema desde a especificação até a implementação do sistema de automação.

No final de cada fase existe uma definição particular do sistema, que são conhecidas como níveis de abstração do sistema, a evolução destes níveis de abstração deve ser controlável de forma a obter um produto final dentro das condições de contorno dadas na especificação do sistema, esta controlabilidade da evolução do níveis de abstração de um sistema permite o gerenciamento dos graus de produtividade, qualidade e custos do projeto [Dewan, 93], [Herbert, 88].

Dentro deste contexto, o nível de abstração definido na fase da arquitetura torna-se a ponte de ligação crítica entre o projeto propriamente dito, e a implementação do sistema, quaisquer tentativa de implementação sem a definição da arquitetura deve produzir fortes impactos no desenvolvimento do projeto [Garlan, 95].

Para conseguir essa evolução gradativa e controlável do projeto, a metodologia ODP apresenta a arquitetura como uma consequência direta da especificação ODP, e como tal representa uma estrutura do sistema constituída de objetos distribuídos e interfaces claramente definidas.

De outro lado, também é bom comentar que existe uma relação direta entre a arquitetura ODP, a especificação ODP e o modelo de referência SAA, como pode se observar na figura 3.5.

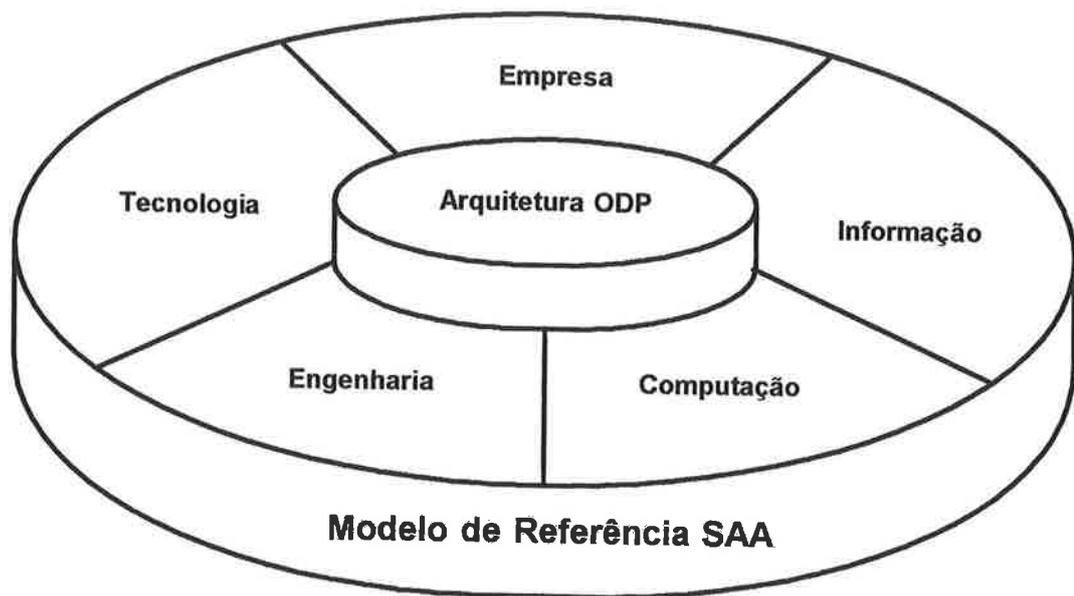


Figura 3.5 Relação entre a Arquitetura ODP, a Especificação ODP, e o Modelo de Referência SAA

Nesta figura, os cinco pontos de vista entorno da arquitetura representam os diversos requisitos a serem considerados; o modelo de referência SAA suportando a arquitetura e os pontos de vista, indica que este modelo define o contexto geral do sistema e a estrutura funcional da arquitetura, segundo o tipo de sistema de automação desejado.

3.4.2 Estrutura da Arquitetura

Este item apresenta o embasamento conceitual do modelo distribuído e apresenta também a estrutura da arquitetura ODP.

3.4.2.1 Modelo Conceitual

A arquitetura proposta baseia-se no modelo conceitual de objetos distribuídos apresentado na figura 3.6, este modelo está constituído de um conjunto de objetos distribuídos que interagem sobre a coordenação de um objeto gerenciador [Nicol, 93]. Os elementos que constituem este modelo conceitual são [Guerroui, 97], [Kythe,96]:

- Objeto cliente: são objetos que utilizam os recursos distribuídos configurados no sistema distribuído e cuja estrutura representa as aplicações específicas distribuídas;
- Objetos recursos: são objetos distribuídos que representam os recursos do sistema e cuja estrutura representa as aplicações, os servidores, os sistemas de base de dados, as ferramentas computacionais, entre outros objetos computacionais;
- Objeto gerenciador: este objeto ou conjunto de objetos permitem o gerenciamento da interação entre os objetos clientes e os objetos recursos. Este objeto tem a capacidade de apresentar de forma transparente os

recursos dos sistemas aos outros objetos distribuídos, sendo que estes recursos podem ser estruturas não orientadas a objetos [Tschicholtz, 96];

- Interfaces distribuídas: são elementos definidos que permitem a troca de informações entre os diferentes objetos distribuídos, estas interfaces comunicam o objeto gerenciador com o objeto cliente e o objeto recurso respectivamente.

O funcionamento dos objetos dentro deste modelo realiza-se da seguinte forma: o objeto cliente utiliza as interfaces disponíveis para se conectar ao objeto gerenciador e pedir um o serviço a um objeto recurso.

Este objeto gerenciador encarrega-se de localizar o recurso, configurar a interface correspondente ao objeto recurso, encaminhar o pedido, esperar a resposta e encaminhar, novamente para objeto cliente.

3.4.2.2 Arquitetura Distribuída ODP

O modelo de arquitetura distribuída do item anterior, representa o embasamento conceitual necessário para definir a arquitetura ODP, que é mostrada na figura 3.7. Esta arquitetura ODP esta constituída dos seguintes elementos [Carlson, 96], [Liang, 94], [Liang, 97]:

- Objeto aplicação: estes elementos representam os objetos automação de cada camada nível do SAA e seguem a especificação do ponto de vista da computação. Na implementação, estes objetos representam as aplicações distribuídas e os recursos no sistema de automação;
- Objeto aplicação ODP: são elementos que representam as aplicações distribuídas no sistema e que foram desenvolvidas utilizando o padrão ODP.

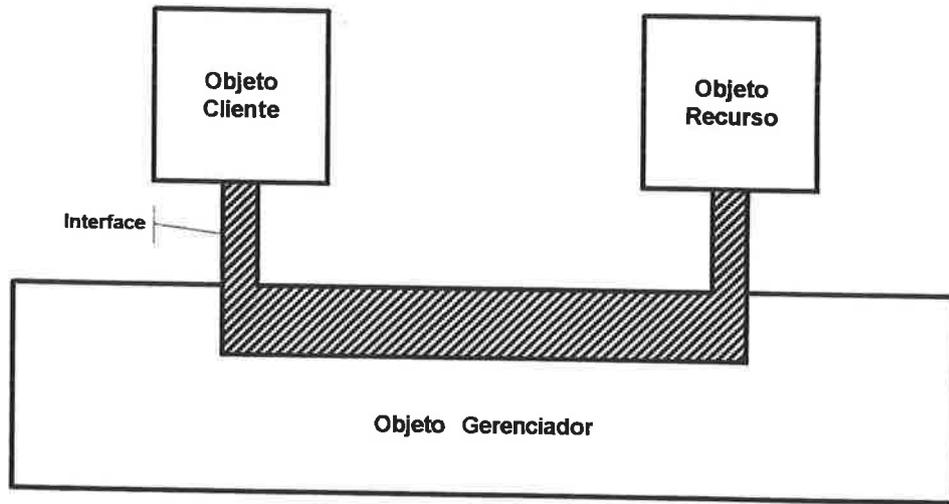


Figura 3.6 Modelo Conceitual da Arquitetura Distribuída

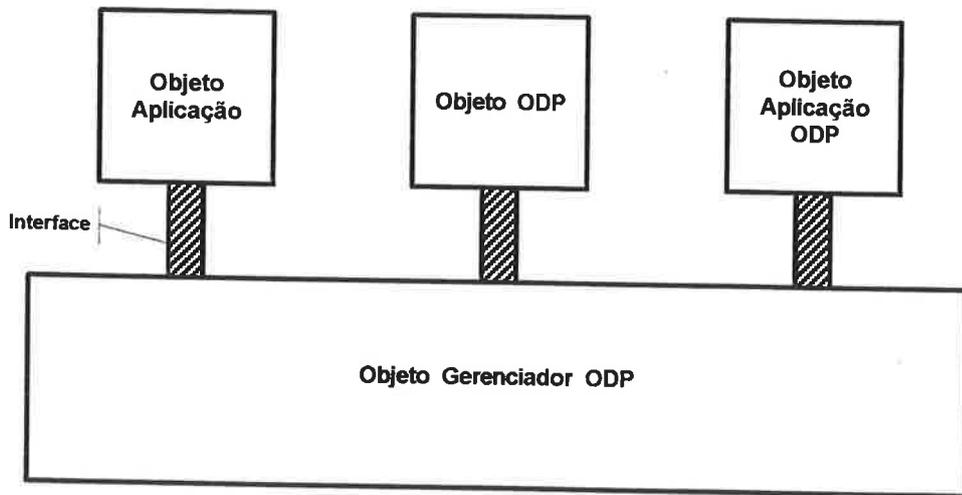


Figura 3.7 Arquitetura ODP

Estas aplicações correspondem às aplicações ODP corporativas de uso geral, e às aplicações ODP específicas, próprias de cada aplicação de automação;

- **Objetos ODP:** estes elementos oferecem serviços que permitem o funcionamento transparente das aplicações distribuídas no sistema. Estes serviços correspondem basicamente as funções ODP, especificadas no item seguinte;
- **Objeto gerenciador ODP:** é o elemento importante da arquitetura ODP, pois permite a interação transparente entre os objetos aplicação e os objetos aplicação ODP, além disso, também utiliza e gerencia a utilização dos objetos ODP. Este objeto vem sendo especificado pelos pontos de vista computacional e de engenharia.

Cabe ressaltar, que os objetos distribuídos interagem através de interfaces corretamente definidas nos pontos de vista e que correspondem ao aspecto do comportamento dos objetos.

Para completar a definição da arquitetura ODP, apresenta-se a seguir a especificação das funções ODP, que são implementadas nos objetos ODP.

3.4.3 Funcionalidades ODP

A funcionalidade ODP representa a especificação das funções ODP básicas que foram definidas na arquitetura ODP.

Esta especificação é um documento que está constituído da definição das funções ODP e das regras das transparências. Este documento é composto na forma de um dicionário de funções e utiliza-se o padrão de especificação [IEEE, 93].

3.4.3.1 Definição das Funções ODP

A especificação das funções ODP está organizada em cinco grupos genéricos denominados: gerenciamento, coordenação, repositórios e segurança.

O grupo genérico de gerenciamento especifica os seguintes serviços:

- Serviços que implementam os mecanismos que permitem o gerenciamento dos nós, das cápsulas e os grupos;
- Serviços que implementam os mecanismos de gerenciamento dos objetos engenharia.

O grupo genérico de coordenação especifica os seguintes serviços:

- Serviços que implementam os mecanismos que manipulam o histórico dos eventos do sistemas e a base de informação dos objetos;
- Serviços que implementam os mecanismos de recuperação, desativação, reativação, copia e migração dos objetos.

O grupo genérico de repositórios especifica os seguintes serviços :

- Serviços que manipulam a base de dados de armazenamento de informações;
- Serviços que gerenciam os repositórios que armazenam os endereços e tipos das interfaces dos objetos engenharia;
- Função de localização de serviços distribuídos (Trading) que está sendo especificado pela norma [ISO, 95e].

O grupo genérico de segurança especifica os seguintes serviços:

- Serviços que implementam os mecanismos de controle de acesso, autenticação e de auditoria;
- Serviços que implementam os mecanismos que suportam a integridade dos dados e os mecanismos de criptografia.

3.4.3.2 Regras das Transparências

As regras das transparências são especificações em linguagem natural que apresentam as restrições e requisitos que devem ser consideradas nas funções ODP. Entre as regras de transparências estão:

- Regras de falhas que definem o esquema de estabilidade que contém os modos de falha que os objeto não devem exibir;
- Regras de migração que definem o esquema de mobilidade que contém restrições no tempo de respostas, desempenho e segurança;
- Regras de persistência que definem o esquema de persistência que contém as restrições das funções de processamento, armazenamento e comunicação;
- Regras de replicação que definem as restrições da função respectivas;

A arquitetura ODP deverá ser definida em cada nível hierárquico do sistema aberto de automação (SAA), e cada objeto desta arquitetura será configurado segundo a funcionalidade de cada nível do SAA.

3.5 Projeto ODP

Esta fase da metodologia ODP é denominada de projeto ODP do sistema, porque engloba as atividades que permitem a definição dos objetos computação que pertencem a cada nível hierárquico do sistema aberto de automação (SAA) e que constituem os elementos da arquitetura ODP.

As atividades realizadas nesta fase são denominadas de análise dos níveis hierárquicos e de projeto detalhado, respectivamente. A primeira atividade visa obter os modelos de objetos e a segunda atividade procura obter a definição dos modelos de processadores correspondente a cada nível do SAA.

Desta forma, deve-se obter ao final desta fase, um conjunto de arquiteturas ODP para cada nível do SAA, interconectados através de objetos gerenciadores ODP.

Cabe ressaltar, que esta fase utiliza os subsídios gerados nos trabalhos de [Tanomaru, 94] e [Matsuyama, 97]. Nestes trabalhos aplicou-se a metodologia orientada a objetos da referência [Coad, 91], que por suas características adequou-se corretamente aos objetivos dos autores.

Neste trabalho, na fase de análise, substitui-se a técnica [Coad, 91] pela técnica do [Rumbaugh, 91], e na fase de projeto utiliza-se os modelos de processadores.

Em termos de diagramas e nomenclatura adotou-se a apresentada pelo [Rumbaugh, 91] e [UML, 97], pois oferece benefícios importantes, como pode-se ver em [Hirama, 95], e porque é a técnica recomendada no padrão ODP, desta forma procura-se manter a coerência no desenvolvimento.

3.5.1 Análise dos níveis hierárquicos

Esta atividade objetiva a determinação do modelo estático, do modelo dinâmico e do modelo funcional de cada domínio do nível hierárquico do sistema aberto de automação (SAA). Para poder atingir os objetivos desta atividade deve-se realizar as seguintes tarefas:

- **Modelo estático:** nesta tarefa deve ser realizada a identificação dos objetos automação, utilizando as informações geradas na especificação dos pontos de vista e na definição do modelo de referência SAA;

Também, dentro desta tarefa, deve-se identificar as associações ou relacionamentos, definir os atributos, e, finalmente, determinar os serviços específicos dos objetos automação. O produto final, desta tarefa, será um diagrama de objetos;

- **Modelo dinâmico:** é objetivo desta tarefa determinar o comportamento dos elementos que compõem os níveis hierárquicos do sistema aberto de automação (SAA). Para atingir esse objetivo, deve-se utilizar a técnica seguinte: preparação dos principais cenários do nível hierárquico, preparação de um diagrama de eventos, e, finalmente, obter como produto final, um diagrama de estados;
- **Modelo funcional:** esta tarefa visa a análise do processo de manipulação da informação nos níveis hierárquicos do SAA. Para atingir este objetivo, utiliza-se como ponto de partida, a informação gerada na especificação do ponto de vista da informação, identificando-se as informações de entradas e saídas, as funções a executar, para finalmente apresentar, como produto final, um diagrama de fluxos de dados.

Recomenda-se, nesta atividade, utilizar ferramentas computacionais, onde serão construídos os diagramas ou modelos e onde será estruturado um repositório do projeto, que será utilizado para controle de versões, tomada de

decisões e garantia da qualidade do projeto. Um exemplo desta ferramenta é apresentado na referência [Rational, 97].

3.5.2 Projeto detalhado

O projeto detalhado dos níveis hierárquicos visa a obtenção de modelos de processadores. Estes modelo estará constituído de módulos de processadores padrão, que por sua vez alocará os objetos automação identificados na análise. Para a realização desta atividade deve-se realizar as seguintes tarefas:

- Identificação dos elementos da arquitetura ODP: aqui deve-se mapear os objetos automação na arquitetura ODP, isto é, identificar quais são os objetos que compõem os objetos aplicação, os objetos aplicação ODP, os objetos gerenciador e, principalmente, determinar o objeto comunicação do modelo de referência SAA. O produto final será uma arquitetura ODP específica;
- Alocação em processadores padrão: os objetos da arquitetura ODP são transformados em módulos de processamento padrão que estarão integrados através de uma via de comunicação padronizada. O produto final será a arquitetura final do sistema.

As atividades da fase de projeto ODP são realizadas num ciclo de vida espiral, existe, portanto, uma retroalimentação, que permite a inclusão de novos objetos automação, e, também, permite o controle da qualidade do desenvolvimento.

3.6 Implementação

A fase denominada de implementação da metodologia ODP, tem por objetivo substituir os elementos da arquitetura do sistema por elementos da tecnologia da informação. A especificação do ponto de vista tecnológico oferece subsídios necessários.

Estes elementos da tecnologia da informação podem ser equipamentos de automação, aplicativos, ferramentas computacionais, gerenciadores de base de dados, redes de comunicação, bibliotecas de programas ou quaisquer elementos que cumprem com a especificação definida na fase anterior.

Cabe ressaltar que estes elementos da tecnologia da informação podem ser sistemas abertos ou sistemas proprietários, dependendo da necessidade do sistema aberto de automação em questão.

Obviamente que esta fase engloba os diferentes tipos de testes como os modulares e os de integração, e, finalmente, é realizada a validação do sistema utilizando como informação básica a especificação ODP.

Este capítulo apresentou, a metodologia de desenvolvimento denominada de ODP, no próximo capítulo apresenta-se um projeto experimental, onde será utilizada esta proposta.

Capítulo 4 Desenvolvimento de um Sistema Aberto de Automação de Projetos

Neste capítulo será apresentado o experimento que utiliza a metodologia proposta no capítulo anterior, e cujo objetivo é demonstrar a aplicação do padrão ODP no desenvolvimento de um sistema aberto de automação.

O conteúdo deste capítulo engloba a definição do experimento, a caracterização do ambiente de projeto dos grandes empreendimentos, cuja estrutura suporta a engenharia simultânea, e os resultados de todas as fases da metodologia ODP.

Finalmente, será apresentada a estrutura do sistema aberto de automação de projetos, no nível de abstração denominado de especificação da implementação.

4.1 Considerações Gerais sobre o Experimento

Neste item apresenta-se as considerações sobre o experimento, no que diz respeito à definição, objetivo, resultados e justificativa do experimento.

No capítulo anterior foi apresentada a estrutura metodológica ODP, cujo embasamento conceitual segue as recomendações do padrão ODP e cuja estrutura possui um ciclo de vida seqüencial que permite um desenvolvimento controlável.

Para analisar a capacidade de aplicação da metodologia ODP, será necessária a criação de um ambiente que permita o treinamento em todos os procedimentos, técnicas e conceitos que esta estrutura engloba.

Este ambiente de experimentação caracteriza-se por estar dentro do contexto de um projeto de sistema aberto de automação, isto é, o desenvolvimento deste projeto de automação torna-se o mecanismo de teste da metodologia ODP.

Os resultados do desenvolvimento serão analisados, com o objetivo de mostrar as conclusões sobre a eficácia desta proposta.

O desenvolvimento do projeto de automação estará orientado para a área de elaboração de projetos, dentro desta área toma-se um exemplo de ambiente de projeto e procurar-se-á uma solução de automação dentro do conceito apresentado em [Martucci, 92].

Este exemplo de ambiente de projeto engloba os projetos dos grandes empreendimentos e utiliza a engenharia simultânea, com o objetivo de aprimorar o controle do tempo de desenvolvimento, o custo e a qualidade do grande empreendimento [Becerra, 96b], [Cutkosky, 93].

O experimento será realizado segundo o esquema seqüencial da metodologia ODP, será executada cada fase da metodologia, e serão encontrados os seguintes produtos específicos: o modelo SAA, os modelos dos pontos de vista, a arquitetura de processadores e a especificação da implementação.

Estes produtos específicos serão analisados de forma a encontrar a relação e o impacto em três aspectos gerais: qualidade, custo e tempo de desenvolvimento. Esta análise gerará resultados que serão de natureza qualitativa, subjetiva e heurística, não envolvendo nenhuma métrica específica.

Justifica-se esta forma de resultados, primeiro, porque a abrangência deste trabalho de tese não engloba a definição de métricas e, segundo, para a determinação de algum tipo de métrica é necessário que a metodologia ODP alcance um determinado grau de maturidade, que envolve sua experimentação

em diferentes tipos de projetos de automação, onde deverão se encontrar resultados que poderão equacionar algum tipo de métrica [Spina, 98].

Com relação a completeza e correção da metodologia ODP, serão colocadas conclusões, mas dentro do contexto da aplicação selecionada, indicando, que a metodologia possui uma grau de completeza inicial, justificada em 3.2 [Senderen, 95], [Jordan, 91].

De qualquer forma, dada a estrutura sistemática da metodologia ODP, pode-se incluir novos elementos e mudanças como consequência dos resultados obtidos em outras aplicações.

Com relação à validade dos modelos, documentos e proposta final do sistema, estes serão avaliados como em todo projeto, comparando com a especificação dos requisitos iniciais, especialmente a especificação dos pontos de vista.

A figura 4.1 apresenta os elementos do experimento, a entrada constituída da estrutura metodológica, o experimento constituído do ciclo metodológico e do projeto SAA e as saídas constituídas dos resultados.

Finalmente, sobre a utilização no experimento de um ambiente com engenharia simultânea aplicado aos projetos dos grandes empreendimentos, justifica-se, primeiro, porque existe um grande interesse na comunidade científica sobre propostas de automação em este tipo de ambiente [Kaplan, 92], [Klein, 93], [Silva, 96], [Kruglianskas, 93].

Uma segunda justificativa está na complexidade dos requisitos, este tipo de ambiente exige a implementação de um sistema de automação, que seja aberto, distribuído e cooperativo, que permita o fluxo contínuo de informação, de forma a integrar os elementos do projeto e manipular um banco de dados único de projeto [Lawson, 93], [Karinthi, 92a], [Karinthi, 92b]. Este tipo de sistema está dentro do escopo dos sistemas ODP.

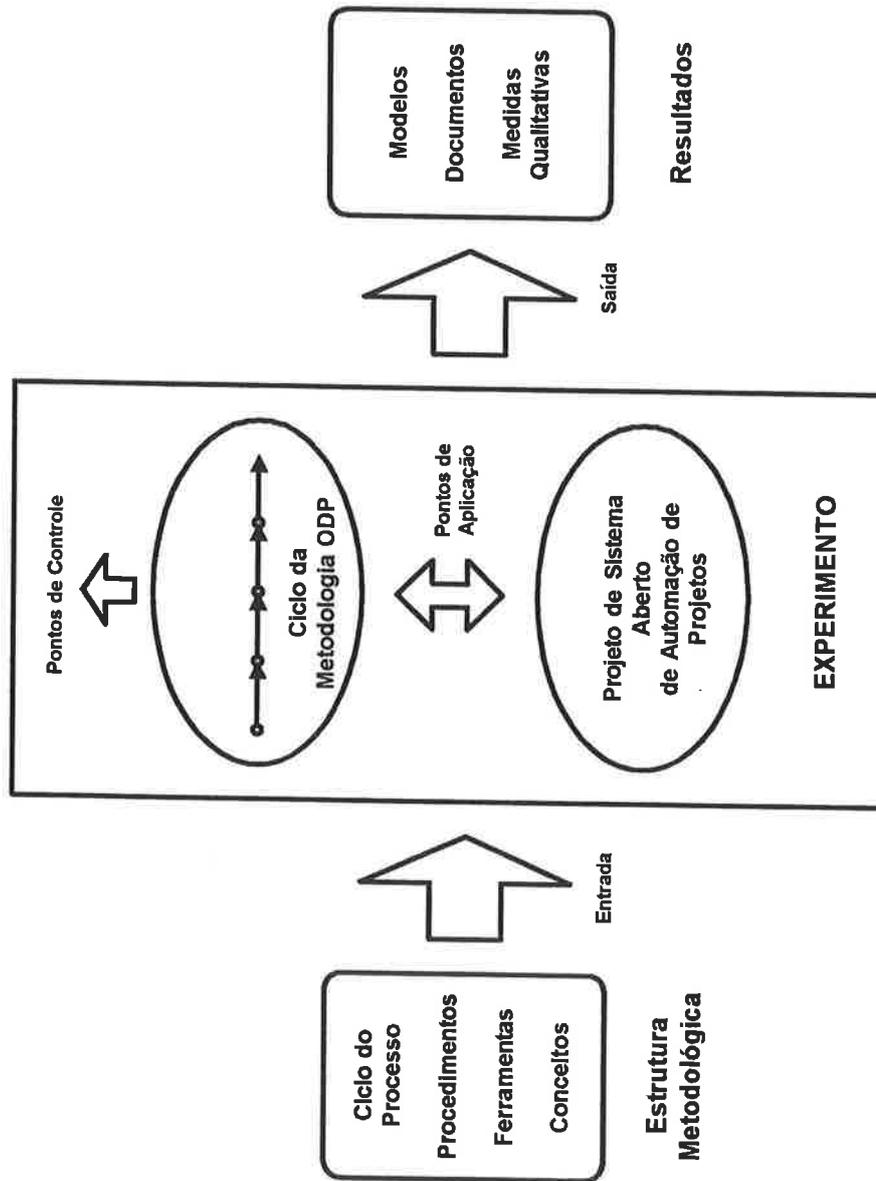


Figura 4.1 Elementos do Experimento

Devido a estas razões, é justificável a utilização deste ambiente como escopo principal da experimentação da metodologia ODP.

A seguir, apresenta-se o ambiente de projeto dos grandes empreendimentos que utiliza a engenharia simultânea, denominado no decorrer deste capítulo como ambiente de projeto.

4.2 Engenharia Simultânea e os Projetos dos Grandes Empreendimentos

O mercado atual possui um comportamento extremamente dinâmico, as variáveis mudam continuamente e exige das corporações a inclusão de novos procedimentos, técnicas, teorias e tecnologias em suas estruturas, com o objetivo de fortalecer seus graus de competitividade, produtividade e qualidade.

Por muitos anos a introdução de graus de automação foi uma das ferramentas mais utilizadas para transformar as corporações em competitivas. Com o mesmo objetivo, as corporações, procuram realizar mudanças na metodologia de desenvolvimento dos produtos ou serviços [Junqueiras, 95], [Gatenby, 94], [Dwivedi, 91].

Dentro deste contexto, a engenharia simultânea ou engenharia concorrente ou engenharia paralela é uma forma de desenvolvimento cuja estrutura possui um alto grau de flexibilidade, permitindo sua evolução e modificação contínua, e visa reduzir os custos, o tempo do desenvolvimento e aumentar a qualidade [Maliniak, 91], [Lawson, 93].

4.2.1 Engenharia Simultânea

O ambiente de projeto que engloba os grandes empreendimentos, é caracterizado por utilizar tempos de desenvolvimentos muito longos e por

manipular orçamentos muito altos, e cuja estrutura envolve muitos processos, equipes, sistemas e ferramentas [Steven, 87], [Berntain, 96].

Neste tipo de projetos os principais problemas estão localizados na geração de inconsistências na informação de projeto, na facilidade com que aumentam os tempos de desenvolvimento, e na falta de um sistema adequado de gerenciamento do projeto como um todo.

Uma forma de solucionar estes problemas e tornar estes projetos produtivos e gerenciáveis está na utilização da engenharia simultânea como metodologia de desenvolvimento.

A engenharia simultânea é uma metodologia de desenvolvimento onde todas as atividades do ciclo de vida são realizadas simultaneamente, com a participação de todos os profissionais de diferentes especialidades desde o início do projeto.

Define-se a engenharia simultânea como o método sistemático aplicado ao desenvolvimento integrado dos serviços ou produtos, enfatizando a importância da satisfação dos requisitos do usuário, incluindo formas de cooperação, responsabilidade e compartilhamento nas equipes, de forma que as decisões de projeto sejam produto de um processo paralelo, sincronizados pelo eficiente sistema de informação, até produzir um consenso [Sprague, 91], [Reddy, 92].

A engenharia simultânea é um método sistemático, porque inclui e integra diferentes tipos de sistemas ao longo do ciclo de vida do desenvolvimento, utilizando o sistema de informação corporativo, desta forma permite-se o trabalho coordenado das pessoas envolvidas nas decisões de projeto.

Em relação à satisfação dos usuários, a engenharia simultânea coloca os requisitos dos usuários, como os critérios mais importantes que deverão ser utilizados na definição e validação das atividades das equipes.

As equipes são os elementos essenciais na engenharia simultânea, devem incorporar formas de cooperação, adotam novas responsabilidades nas decisões do projeto e usam mecanismos de compartilhamento da informação. A forma de trabalho destas equipes deve ser entendida, suportada e seguida por todos os membros da corporação.

As decisões de projeto são produtos importantes que são elaboradas com a participação de todos os especialistas, clientes e pessoas que possuem uma relação direta com os produtos ou serviços fornecidos pela corporação.

As decisões são produtos de processos ou trabalhos paralelos, característica básica da engenharia simultânea, estes processos devem ser corretamente planejados e sincronizados, através de um sistema de informação eficiente, para evitar inconsistências entre os critérios dos membros da equipe.

Além disso, as decisões de projeto tornam-se um consenso da equipe, não é uma decisão de um membro da equipe, desta forma, aumenta-se e distribui-se as responsabilidades sobre o projeto, sobre todos os membros da equipe [Mulvihill, 93].

Finalmente, a utilização da engenharia simultânea traz no ambiente de projeto, os seguintes benefícios, como pode ser verificado em [Dewan, 93.], [Handfield, 94], [Ramana, 93], [Silva, 96]:

- Redução do tempo de desenvolvimento do serviço ou produto, por causa do paralelismo das atividades;
- Homogeneização e melhora da qualidade, originado pela revisão contínua de inconsistências e erros e dos requisitos do usuário. Esta revisão é realizada por diferentes especialistas, representantes de todas as fases do projeto;

- Diminuição das alterações e dos reprojotos, por causa do trabalho cooperativo e multidisciplinar, os diferentes critérios técnicos relacionados com o produto ou serviço são analisados e incorporados desde o início do projeto;
- Redução dos custos do projeto, é uma conseqüência direta dos benefícios atingidos anteriormente.

Estes benefícios poderão ser atingidos, desde que a engenharia simultânea seja implantada de forma organizada e sistemática, como pode ser visto em [Silva, 96], onde se apresenta uma estratégia de implantação.

É bom indicar que a implantação inicial da engenharia simultânea poderá evoluir segundo o modelo de maturidade CMM (Capability Maturity Model) [Spina, 98].

4.2.2 Ciclo de Vida Paralelo e a Equipe Virtual

Existem dois elementos importantes na estrutura do ambiente de projeto, um desses elementos é o ciclo de vida paralelo de desenvolvimento e o outro elemento é a equipe de desenvolvimento.

O ambiente de projeto possui um ciclo de vida ou processo de desenvolvimento paralelo ou simultâneo, isto é, todos os membros da corporação participam, do desenvolvimento do serviço ou produto, desde a concepção até a fase de suporte.

Para explicar este processo, analisa-se o processo seqüencial apresentado na figura 4.2.a. Neste diagrama, os blocos representam as diferentes fases do processo, sua execução gera produtos que ativam a fase seguinte, é a principal

característica deste tipo de processo. Cabe ressaltar que as atividades de projeto são próprias de cada fase e são independentes das outras fases.

As informações de projeto estão armazenadas em bases de dados específicas de cada fase, estas informações são disponibilizadas, exclusivamente, para os membros da equipe dessa fase, e poderão eventualmente, com prévia autorização, serem utilizadas por outros membros de outras equipes.

Com relação a gerência do projeto, existe uma gerência específica e independente para cada fase do processo, as atividades das diversas gerências são executadas seqüencialmente.

Em relação ao sistema de automação e informação, este possui requisitos técnicos mais exigentes a nível de inter-processo (interior das fases) e menos exigente a nível intra-processo (entre fases).

A figura 4.2.b apresenta o ciclo de vida do ambiente de projeto, este diagrama possui as seguintes características:

- A execução das fases do projeto são realizadas de forma paralela, existe um trabalho cooperativo entre todas as fases e, como consequência desta forma de trabalho, são gerados produtos com adequados graus de qualidade [Reddy, 92], [Palmer, 91];
- As informações de projeto são armazenadas numa base de dados única, e são disponibilizadas da mesma forma para todos os membros das equipes incluindo as gerências;
- A gerência deixa de ser exclusiva de cada fase e torna-se uma gerência geral, onde as atividades de coordenação são fatores críticos;

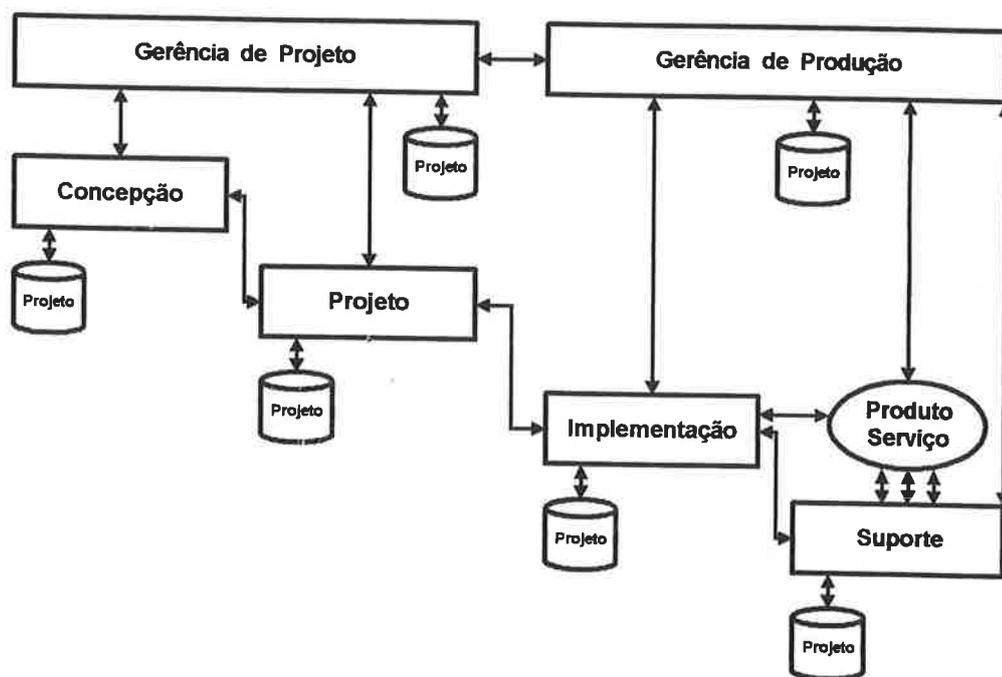


Figura 4.2 a Ciclo de Vida Sequencial

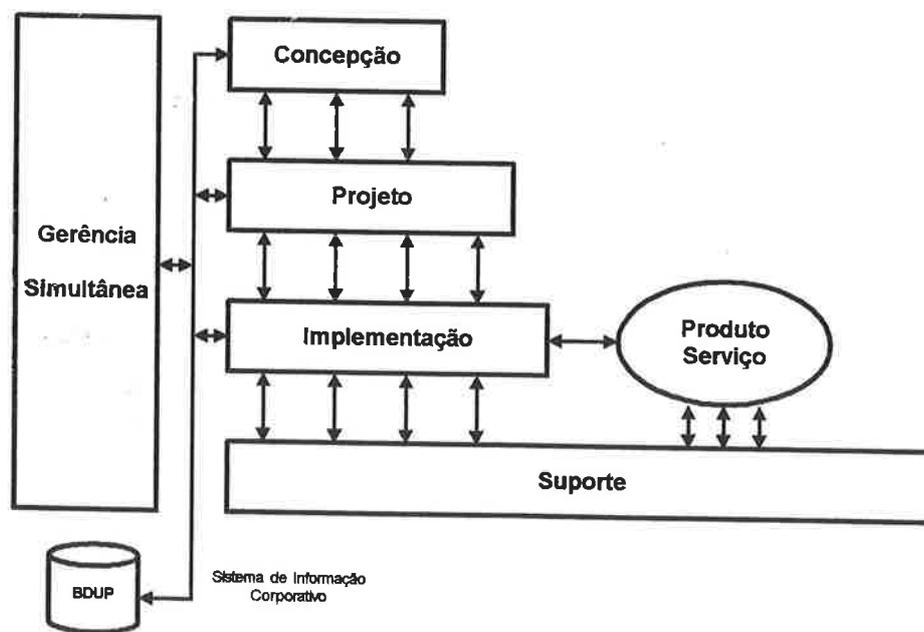


Figura 4.2 b Ciclo de Vida da Engenharia Simultânea

- O sistema de automação e informação incluirá novas características em seus requisitos técnicos, especialmente, para permitir a interoperabilidade de todos seus elementos, com o objetivo de suportar um ambiente cooperativo que permita a manipulação eficiente da informação de projeto por todos os elementos da corporação.

Outro elemento importante neste tipo de ambiente de projeto é a equipe de desenvolvimento, existe uma equipe multidisciplinar constituída de elementos com diferentes perfis profissionais trabalhando cooperativamente.

Esta equipe do ambiente de projeto, na realidade, está composta de muitas equipes, que por sua vez, estão constituídas de muitos profissionais de diferentes áreas, distribuídos fisicamente na corporação e integrados através do sistema de informação.

A eficiência do sistema de informação, onde existe um fluxo contínuo de informação de projeto, permite que a complexidade organizacional das equipes sejam observadas como uma entidade coerente, coesa e abstrata, denominada de equipe virtual [Dewan, 93], [Malianiak, 91].

A equipe virtual é uma definição conceitual, cuja implantação deve ser estruturada adequadamente na corporação, caso contrario, poder-se-á reduzir as possibilidades de obter os benefícios da engenharia simultânea.

4.2.3 Base de Dados Única de Projeto

As atividades de projeto realizadas pela equipe virtual, atividades de desenvolvimento ou gerenciais, estão caracterizadas por gerar uma forte interação com o sistema de informação da corporação, pois o elemento básico de todas as transações cooperativas são as informações de projeto.

A manipulação das informações de projeto exige do gerenciador das informações requisitos importantes, como por exemplo, formatação adequada dos dados, gerenciamento das transações, formas de consistência de dados, e mecanismos de segurança, entre outros aspectos [Benford, 92], [Berra, 90], [Blair, 94].

Dentro deste contexto, qualquer execução das transações cooperativas fora dos requisitos especificados aumentariam o número de inconsistências na informação do projeto, e, como consequência, inviabilizaria o projeto como um todo.

Assim, uma forma de diminuir os riscos de erros no projeto, por causa destas inconsistências na informação de projeto, pode ser atingida através da definição de um modelo conceitual de dados, denominada de base de dados única de projeto.

A base de dados única de projeto, é uma estrutura abstrata de dados que representa todas as complexidades de dados referentes ao projeto, e está disponível a todos os membros da equipe virtual, como é mostrado na figura 4.3, onde a base única de projeto é o núcleo central das transações cooperativas.

A estrutura lógica da base única de projeto está composta dos seguintes tipos de informação [Chang, 93], [Kamel, 92]:

- Informações referentes ao produto: correspondem às informações técnicas dos produtos e serviços desenvolvidos na corporação. Estas informações estão composta de especificações dos requisitos, definições de projeto (*Design*), diferentes formas de modelos, diversas arquiteturas, especificações técnicas sobre tecnologia e materiais, considerações de segurança, planos de testes, entre outros relativos às características do produto ou serviço no desenvolvimento;

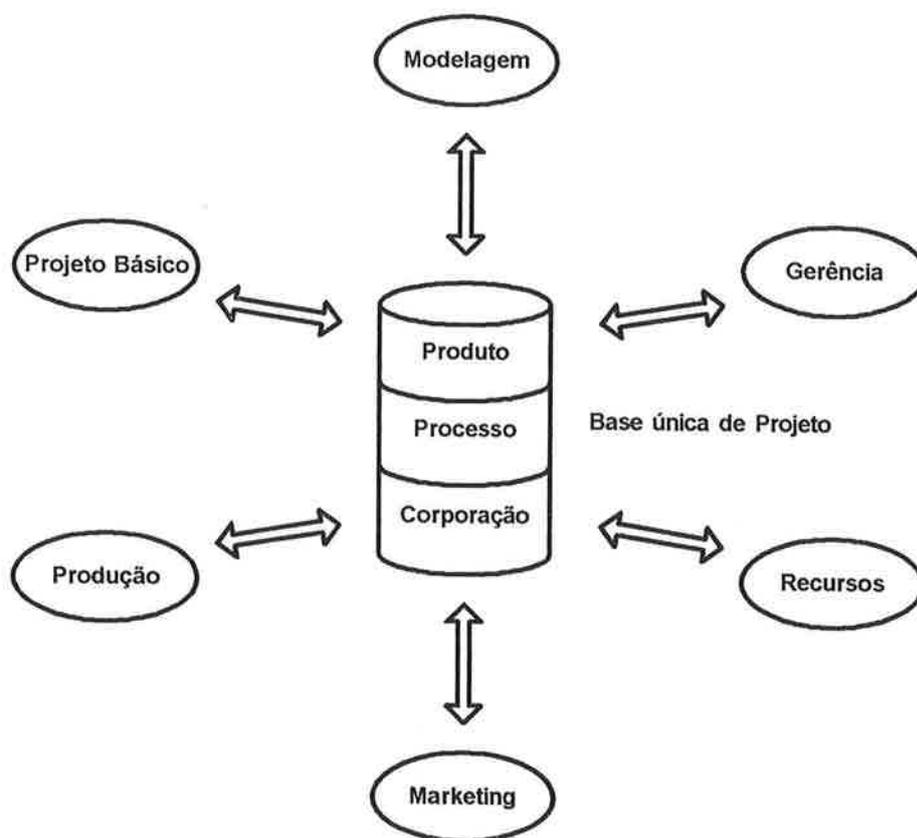


Figura 4.3 Modelo Conceitual de Base de Dados única de Projeto

- Informações referentes ao processo: correspondem às definições dos processos que são aplicados no desenvolvimento dos produtos ou serviços. Estas informações englobam os fluxos de trabalhos e *workflows*, estados do processo, requisitos de desenvolvimento, definição de atividades e funções, regras de interação, formas de coordenação, controle de versões, usuários envolvidos, entre outras informações relativas à forma de desenvolver um produto ou serviço;
- Informações referentes à empresa: correspondem às informações referentes à estrutura de suporte do desenvolvimento. Estas informações envolvem os recursos humanos, as ferramentas de apoio ao projeto, planejamentos, planos estratégicos, normas corporativas, documentos de decisões, distribuição de trabalho, históricos de produtos e serviços, entre outras informações relativas aonde está sendo realizado o desenvolvimento.

A estrutura física da base única de projeto poderá ser implementada segundo os diferentes modelos apresentados em [Nankman, 96]. Para este ambiente de projeto, será utilizado o modelo distribuído como é apresentado na figura 4.4.

Neste modelo, a base de dados única de projeto é interligada às bases setoriais, que por sua vez estão conectadas as bases locais, em cada nível de interação existe um mecanismo de controle de acesso e versão para manter a consistência das informações [Schill, 91], [Sriram, 92].

As bases setoriais possuem uma cópia parcial das informações da base única, e as bases locais possuem copias da base setorial, que são utilizadas nas atividades específicas dos usuários. A base local pode ser centralizada e compartilhada pelos diversos usuários desse grupo.

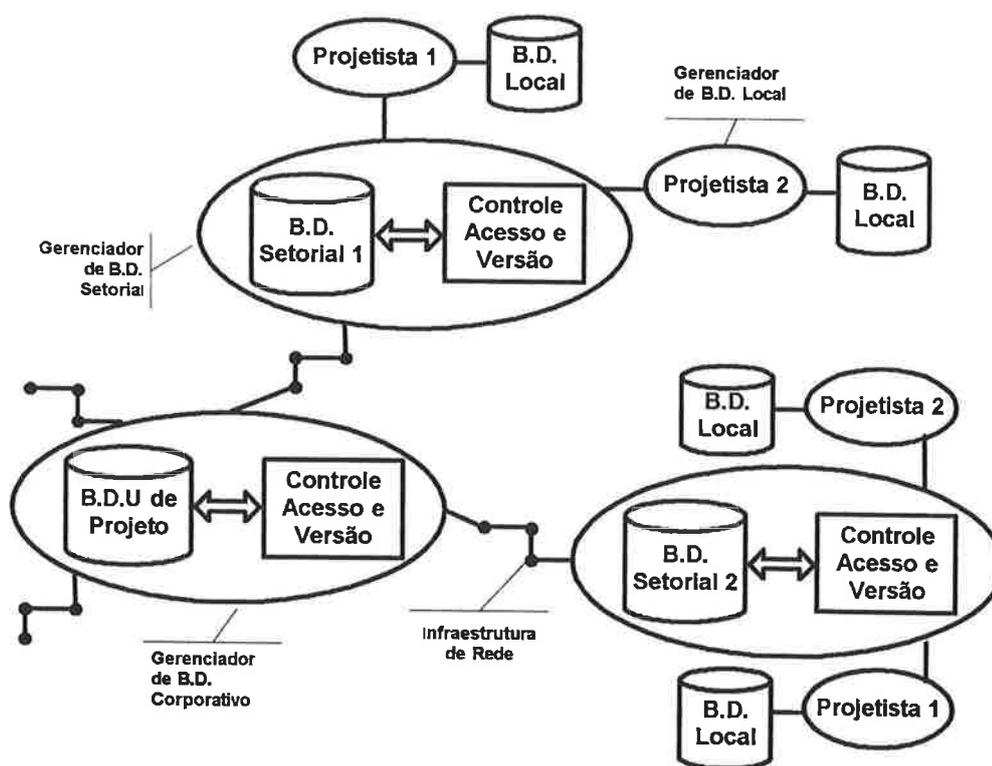


Figura 4.4 Modelo Físico da Base de Dados Única do Ambiente de Projeto

Os blocos de controle de acesso e versões, das diferentes bases, realizam cópias automáticas ou segundo a necessidade do cliente, das informações atualizadas e executam as funções de coordenação entre base de dados, para evitar a inconsistência dos dados.

Cabe indicar, com relação às funções do gerenciador de bases de dados, que existem outras funções inerentes, como pode-se ver em [Nankman, 96], mas, por causa deste ambiente de projeto, foi ressaltada apenas a função de controle de acesso e versão.

4.2.4 Serviços do Ambiente de Projeto

As diversas tarefas realizadas pela equipe virtual, no ambiente de projeto, são definidas e suportadas pelos denominados serviços. Estes serviços englobam atividades e funções dos elementos envolvidos no ambiente.

Os serviços são identificados no ambiente cooperativo, utilizando o modelo mostrado na figura 4.5, que representa uma modificação da proposta apresentada na referência [Ramana, 93]. Os principais serviços são:

- **Serviços globais:** correspondem às grandes atividades realizadas pela equipe virtual, que são: gerenciamento, planejamento, desenvolvimento, implementação e otimização;
- **Serviços específicos:** correspondem às atividades específicas realizadas pelos membros da equipe, e que correspondem as tarefas de consultar, processar, comunicar, negociar, decidir e arquivar. Na figura 4,5 é mostrado este serviço como suporte dos serviços gerais;

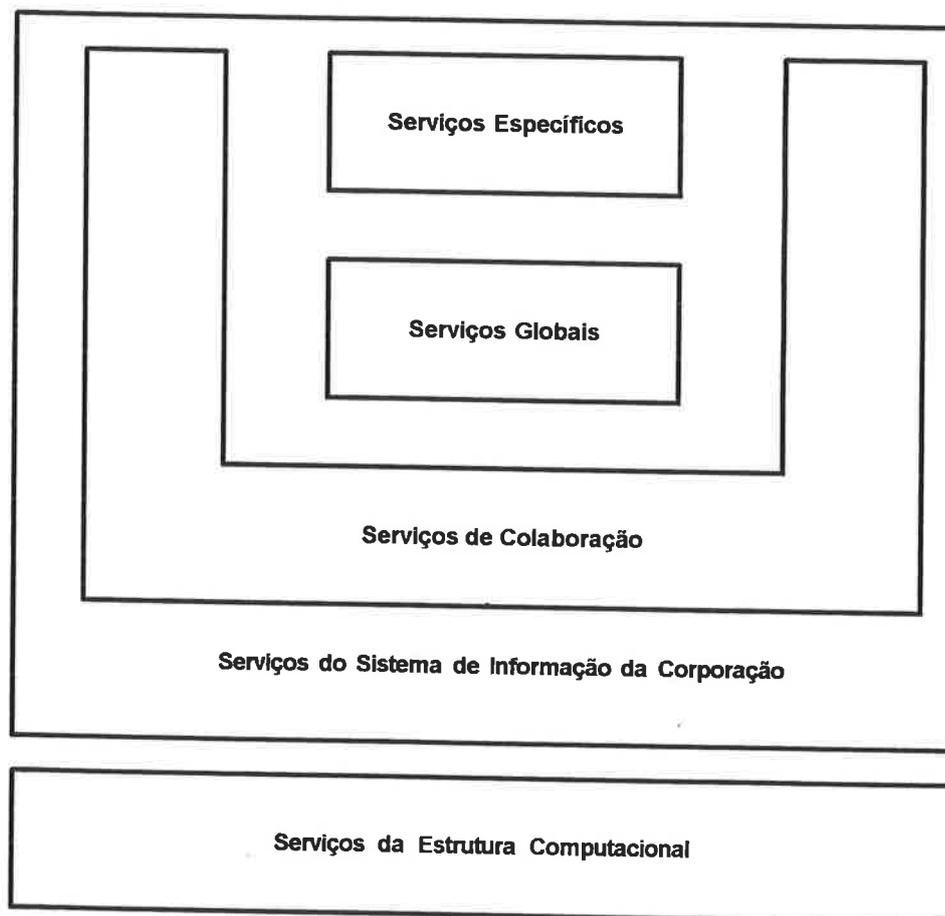


Figura 4.5 Modelo dos Serviços do Ambiente de Projeto

- Serviços de colaboração: conjunto de serviços muito importante, e que correspondem aos serviços de suporte dos blocos anteriores, a figura 4.5 mostra este serviço envolvendo os blocos anteriores. Os serviços a implantar são: intercomunicação, coordenação, compartilhamento das informações e acesso ao histórico do corporativo.

A intercomunicação refere-se as funções de comunicação entre os elementos e aplicativos do ambiente cooperativo; tais como: disponibilizar as reuniões virtuais, como vídeo conferência, facilidades para a manipulação de aplicações distribuídas, como o serviço *trading* ou páginas amarelas para identificar recursos na corporação [Wolisz, 93].

A coordenação engloba os serviços referentes a formação de equipes, planejamento, gerenciamento, controle de decisões de projeto, controle do processos e controle dos requisitos.

O compartilhamento da informação engloba os serviços de codificação dos dados em formatos e modelos padrão, como por exemplo, o PDES (Product data Exchange), EDI (Electronic data exchange), e o ODA (Open document architecture), [Turban, 98], [Kilov, 95]; serviços de manipulação das diferentes bases de dados e serviço de controle de versões.

O acesso ao histórico da corporação engloba funções que permitam a manipulação das informações referentes ao processo de decisão e de desenvolvimento, como atas de reuniões e documentos técnicos;

- Serviço do sistema de informação corporativo: corresponde a todas as funções relacionadas com a manipulação da base única de dados. Como é mostrado na figura 4.5 este serviço respalda os outros serviços anteriores;

- Serviço da estrutura computacional: corresponde às funções do sistema de rede e do sistema computacional distribuído. Este serviço é o principal suporte de todos os serviços definidos no ambiente de projeto.

4.2.5 Organização

A estrutura organizacional da corporação é um aspecto importante que pode determinar o sucesso da implantação da engenharia simultânea [Labauve, 92].

O ponto central de uma estrutura organizacional, que suporta engenharia simultânea, está na equipe de desenvolvimento, cuja composição abrange os diferentes departamentos da corporação [Karandikar, 92].

A figura 4. 6 apresenta a organização do ambiente de projeto. Iniciando a descrição funcional no nível inferior, encontra-se a equipe de desenvolvimento, constituída das diferentes especialidades, como por exemplo, engenharia de projeto, engenharia da produção, engenharia de sistemas, engenharia da especificação, inclusive, clientes e fornecedores.

O número de representantes de cada especialidade na equipe depende do projeto ou da fase do projeto que se está executando, por exemplo, quando se trata de um projeto de software que está na fase de implementação, a equipe terá uma forte participação de programadores, com um número mínimo dos outros especialistas.

Todas as equipes de desenvolvimento possuem um líder de projeto, figura importante, dedicando-se à coordenação, controle e supervisão do grupo. Esta figura deve ter conhecimentos suficientes, para entender sobre especificação, projeto, produção, integração, teste, validação e suporte.

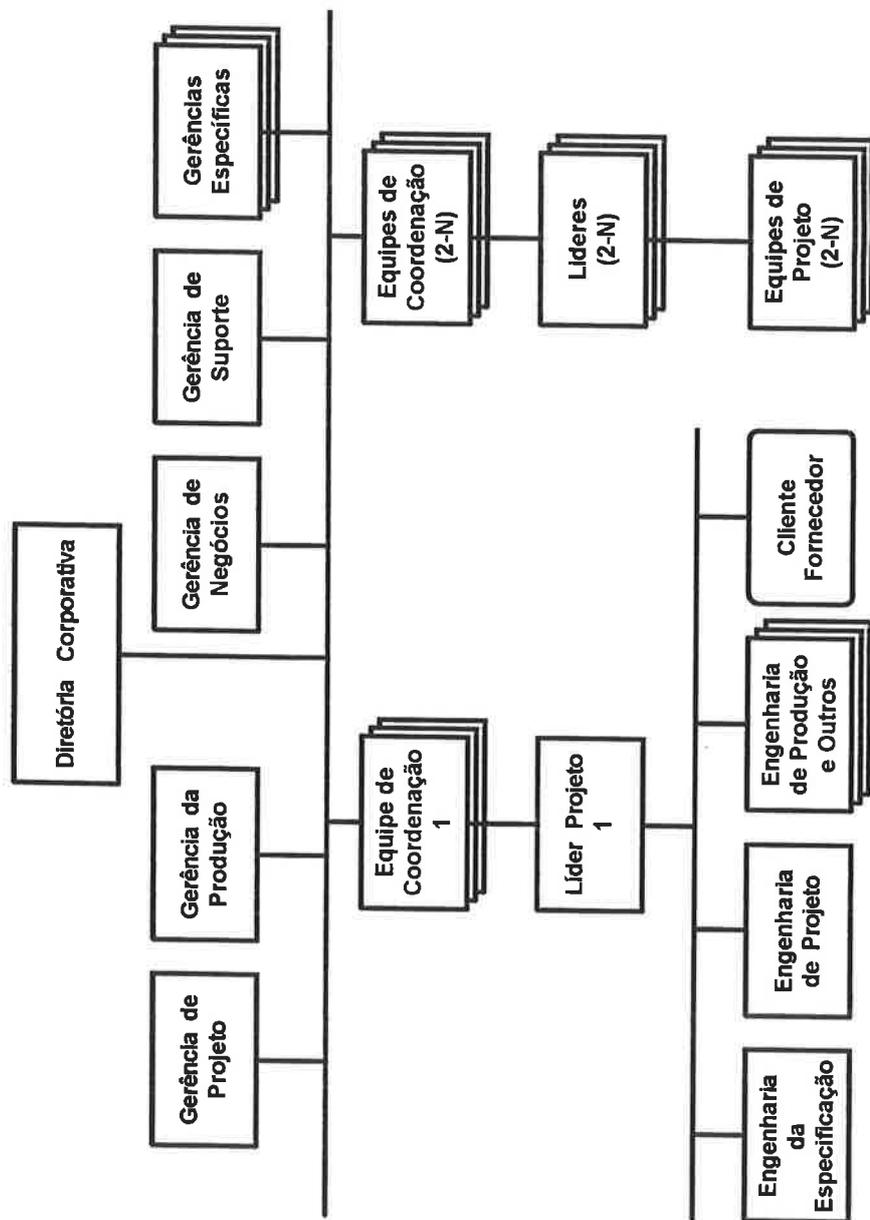


Figura 4.6 Organização de um Ambiente de Projeto

No centro da figura 4.6, estão as equipes de coordenação de projetos, constituída de vários especialistas, cuja missão é coordenar a ação de todos os projetos da corporação.

No nível superior da coordenação de projetos estão os diferentes departamentos envolvidos na corporação, os gerentes destes departamentos, encarregam-se de supervisionar e dar suporte a seus especialistas em cada projeto da corporação.

No topo superior está a diretoria corporativa cujos membros encarregam-se das estratégias, políticas e assuntos de alto nível na corporação.

Tomando como referência qualquer projeto da corporação, a equipe virtual de projeto está constituída por todos os elementos envolvidos no projeto específico, desde a equipe de desenvolvimento até os membros da diretoria corporativa.

4.2.6 Requisitos do Sistema

Finalmente, o ambiente de projeto deverá possuir uma série de requisitos do sistema, definidos como características importantes que devem ser consideradas ao longo do desenvolvimento do sistema de automação e implantadas no sistema final. Estes requisitos são:

- Ambiente cooperativo: este requisito diz respeito a formação de um sistema computacional dentro do contexto do *Groupware* ou ambiente cooperativo, cuja estrutura permite a iteração cooperativa entre seus elementos [Bentley, 94];
- Sistema aberto: requisito que impulsiona a utilização de padrões e normas em diversos aspectos do ambiente de projeto, com objetivo de gerenciar a

heterogeneidade do sistema. Também serão aceitos propostas proprietárias, quando forem necessárias [Becerra, 93];

- Interoperabilidade: deverão ser implantados altos graus de interoperabilidade entre todos os elementos do sistema. Este requisito é importante para poder suportar diversas formas de interação cooperativa, por exemplo, a integração com outros sistemas dos clientes ou dos fornecedores [Becerra, 96b];
- Tempo real: serão implantados considerações sobre tempo real, onde seja necessário, por exemplo, em algumas tarefas referentes à intercomunicação, transações cooperativas e manipulação de informação [Adler, 95];
- Modularidade: o sistema deverá possuir uma estrutura modular, cada módulo envolve um ou um conjunto de objetos automação;
- Segurança e confiabilidade, disponibilidade: estes requisitos serão implantados em diferentes graus de exigência, e segundo os níveis hierárquicos do sistema de automação do ambiente de projeto;

Estes requisitos de projeto e a caracterização do ambiente de projeto dos itens anteriores, definem a complexidade do sistema de automação do ambiente de projeto dos grandes empreendimentos, e justificam a utilização de uma metodologia de projeto adequada.

4.3 Modelo Hierárquico do Ambiente de Projeto

Após a caracterização do ambiente de projeto, a primeira fase da metodologia ODP a desenvolver, é a determinação do modelo hierárquico ou definição do sistema aberto de automação de projeto, cuja estrutura baseia-se no modelo

de referência do item 3.1. A figura 4.7 apresenta o sistema com três níveis: coordenação, controle e instrumentação.

Antes de realizar a análise do sistema de automação, existem duas considerações a ressaltar e que foram colocadas com o objetivo de reduzir a complexidade desta análise, mas sem perder a essência desta fase.

Primeiro, será considerada na corporação apenas uma equipe virtual e um projeto da corporação, este projeto estará constituído do processo de desenvolvimento e do processo de produção, este último referente a implementação do produto ou serviço.

Segundo, a composição dos objetos automação seguirá a proposta [Tanomaru, 94] e estarão compostos de módulos integrados através de uma via de comunicação padrão. Os módulos básicos são: processador, comunicação, interface homem-máquina e gerenciador de base de dados.

A estrutura do nível de coordenação do sistema de automação de projeto está constituída dos seguintes objetos automação:

- Gerenciador da base de dados central: encarregado de oferecer os serviços de gerenciamento e manipulação da base única de projeto;
- Gerenciador da operação: encarregado da coordenação e supervisão da operação dos elementos do nível de coordenação;
- Gerenciador da configuração: permite a localização, manipulação e transferência da configuração de cada elemento do sistema de automação;
- Gerenciador da comunicação: encarregado de gerenciar os serviços de comunicação entre os membros da equipe virtual;

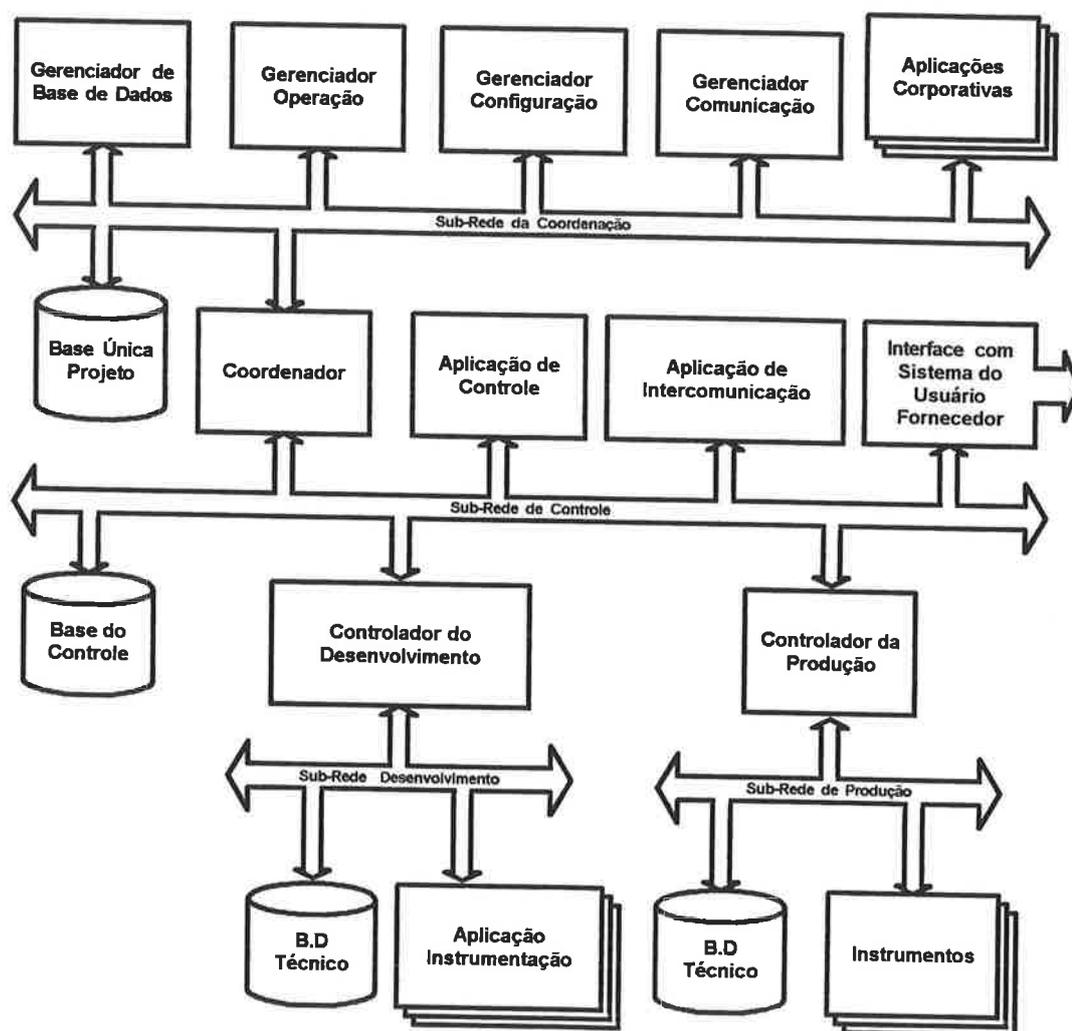


Figura 4.7 Modelo Hierárquico

- Interface homem-máquina: oferece os serviços de interface homem-máquina a nível central para dar apoio ao gerenciamento de todos os elementos do sistema de automação;
- Aplicações corporativas: são serviços específicos dos elementos do nível de coordenação, como o gerenciador da engenharia de produção e manufatura, gerenciador da engenharia de projeto, gerenciador de manutenção, gerenciador de marketing, entre outros [Cerc, 93];

A estrutura do nível de controle e instrumentação do sistema de automação, está constituída dos seguintes objetos automação:

- Coordenador: encarregado de supervisionar e coordenar a operação dos elementos do nível de controle e instrumentação, e manipula a interface homem-máquina a nível sectorial. Em geral, este objeto gerência todos os projetos que estão sobre sua autoridade [Londono, 92], [Kilov, 95];
- Aplicações de controle de projeto: englobam os serviços de apoio ao gerenciamento dos projetos. Estes serviços suportam o planejamento dos projetos, o controle de versões, o controle dos requisitos, o controle de tarefas, o controle de decisões, entre outros [Couison, 95];
- Aplicação da intercomunicação: implanta serviços de comunicação entre os membros da equipe dos níveis de controle e instrumentação. Estes serviços são vídeo conferências, correios eletrônicos, e ambientes para diálogos eletrônicos [Rodden, 92]
- Controlador de desenvolvimento: encarregado da supervisão, controle e coordenação, a nível local, dos elementos que constituem a sub-rede de desenvolvimento ;

- Controlador da produção: encarregado da supervisão, controle e coordenação, a nível local, dos elementos da sub-rede de produção;
- Aplicação da instrumentação: constituídos dos serviços oferecidos pelas aplicações específicas de cada fase do projeto. São as ferramentas computacionais de apoio ao desenvolvimento dos projetos;
- Instrumentos: equipamentos específicos da automação, como equipamentos da célula flexível (manufatura), ambiente de produção de um sistema de informação (software), estrutura do sistema metroviário (serviço), entre outros elementos próprios do escopo da corporação;

De outro lado, no sistema aberto de automação de projeto identificam-se quatro objetos comunicação, que definem a estrutura das seguintes sub-redes de comunicação: a sub-rede da coordenação, a sub-rede de controle, a sub-rede do desenvolvimento e a sub-rede da produção.

Estas redes de comunicação, de alta velocidade, implantam protocolos padronizados das arquiteturas TCP-IP, OSI, ou FieldBus, e protocolos proprietários em alguns segmentos da rede. As sub-redes heterogêneas são interconectadas através de elementos denominados de interfaces de comunicação ou *Gateways*.

Cabe ressaltar, que a sub-rede de controle está interligada à rede dos clientes e dos fornecedores, para efetivar a participação direta destes elementos no processo de desenvolvimento.

Finalmente, pode-se concluir que o sistema aberto de automação de projetos é um sistema distribuído, constituído de objetos automação heterogêneos interligados através de objetos de comunicação, cujo atributo principal é a alta velocidade.

Estas informações do sistema de automação e as informações da estrutura do ambiente de projeto, do item anterior, constituem o principal subsídio para definir os cinco pontos de vista do sistema.

4.4 Modelos dos Cinco Ponto de Vista

Os modelos dos cinco pontos de vistas do ambiente de projeto, que a seguir serão apresentados, resultam da aplicação direta do método definido no item 3.3, denominado de especificação ODP. Utiliza-se a tabela 3.1 como apoio neste processo da modelagem.

Os modelos e diagramas representam a transformação da informação da caracterização do ambiente de projeto, representado em linguagem natural e em modelos abstratos para uma informação mais adequada, para poder tornar a atividade de definição da arquitetura uma tarefa mais simples e coerente.

Cabe ressaltar, que os modelos dos pontos de vista definem o sistema em cinco visões, de forma a aumentar a completeza da especificação.

Neste item, serão apresentados os modelos e diagramas em um nível alto de abstração, com exemplos básicos, importantes e necessários para poder mostrar a técnica. Todos os modelos e diagramas, em detalhe, fazem parte do documento de projeto denominado de especificação ODP.

4.4.1 Modelo da Empresa

A figura 4.8 mostra, o diagrama de objeto do ponto de vista da empresa, que está composto com os seguintes elementos:

- O objeto empresa - equipe da corporação: representa os elementos da equipe virtual que pertencem à corporação. A associação, representada

pelo losango preto, indica que todos os objetos conectados, estarão ativos quando existir um relacionamento cooperativo;

- O objeto empresa - equipe externa: representa a participação do objeto cliente e do objeto fornecedor;
- O objeto empresa - sistema de informação: representa o sistema de informação corporativo, este objeto se encontra ligado aos outros objetos através de uma associação ternária, cuja existência depende da ativação simultânea dos três objetos.

A figura 4.9 mostra um diagrama *Use Case*, que serve para definir uma regra de interação do ponto de vista da empresa. Este diagrama mostra os eventos (*use case*) que ocorrem quando o ator utiliza o sistema.

Este *Use Case* apresenta a relação do líder e do desenvolvedor, quando utilizam o sistema de informação para poder realizar uma negociação de um requisito de projeto.

Os desenhos dos homens representam os atores, o quadro representa o sistema específico, as linhas são estímulos ou informações e as elipses são os eventos.

Como conclusão da análise destes modelos, do diagrama de objetos e do *Use Case*, indica-se que todos os processos e requisitos do ambiente de projeto estão sob o contexto de um forte relacionamento entre a equipe virtual e a base de dados única do projeto. As regras de empresa são definidas segundo este contexto.

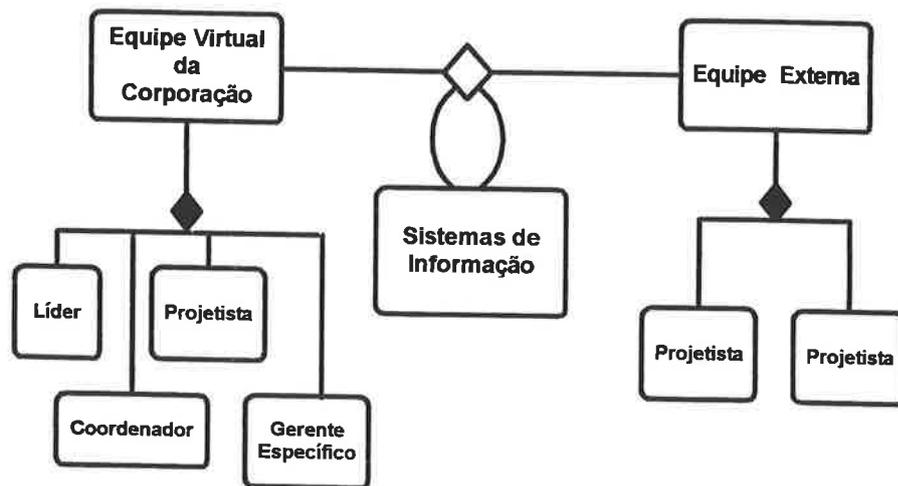


Figura 4.8 Diagrama de Objetos do Ponto de Vista da Empresa

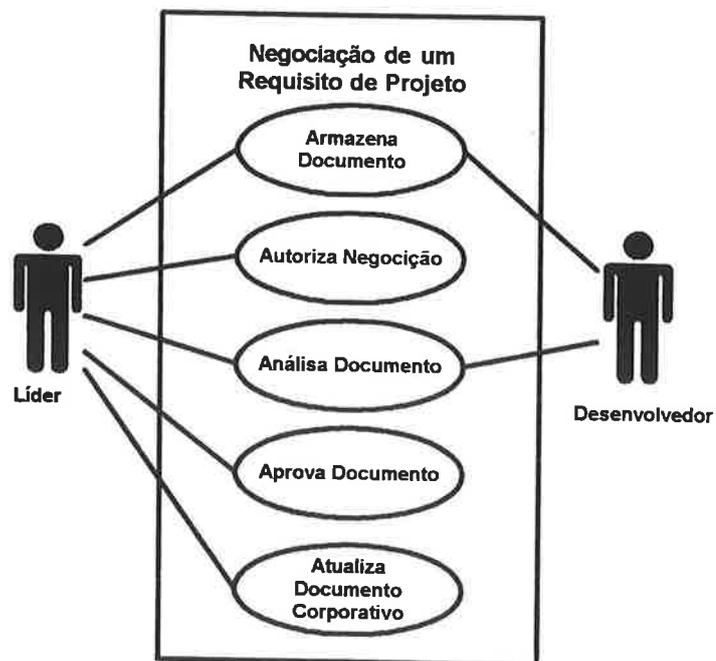


Figura 4.9 Uses Cases de uma Regra da Empresa

4.4.2 Modelo da Informação

Na modelagem da informação são gerados três diagramas, um diagrama de objetos para o esquema invariante, um diagrama de colaboração para o esquema estático e um diagrama de estados para o esquema dinâmico.

A figura 4. 10 mostra, um diagrama de objetos informação, onde o objeto informações de projeto possui uma estrutura hierárquica constituída de objetos produto, processo e empresa (base única de projeto). Esta associação indica que estes objetos são mutuamente exclusivos.

A figura 4.11 apresenta, o diagrama de colaboração de um processo estático da informação, denominado de transformação de um relatório em documento técnico de projeto.

As linhas de associação representam a ligação entre objetos empresa, as flechas representam as mensagens trocadas entre objetos e os blocos representam os objetos empresa.

A figura 4.11 indica que um relatório pode-se transformar em documento técnico, quando existe uma troca de mensagens entre os objetos gerenciador, o objeto estação líder e o objeto sistema.

A figura 4.12 mostra, o diagrama de estado de um esquema dinâmico, denominado de consenso de uma decisão de projeto.

Este diagrama define os estados do objeto informação consenso, para se tornar um objeto informação decisão de projeto.

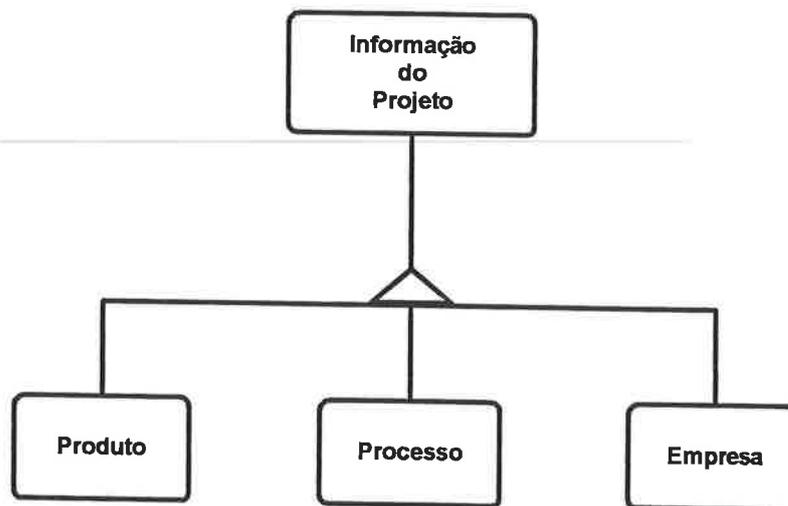


Figura 4.10 Diagrama da Informação do Esquema Invariante

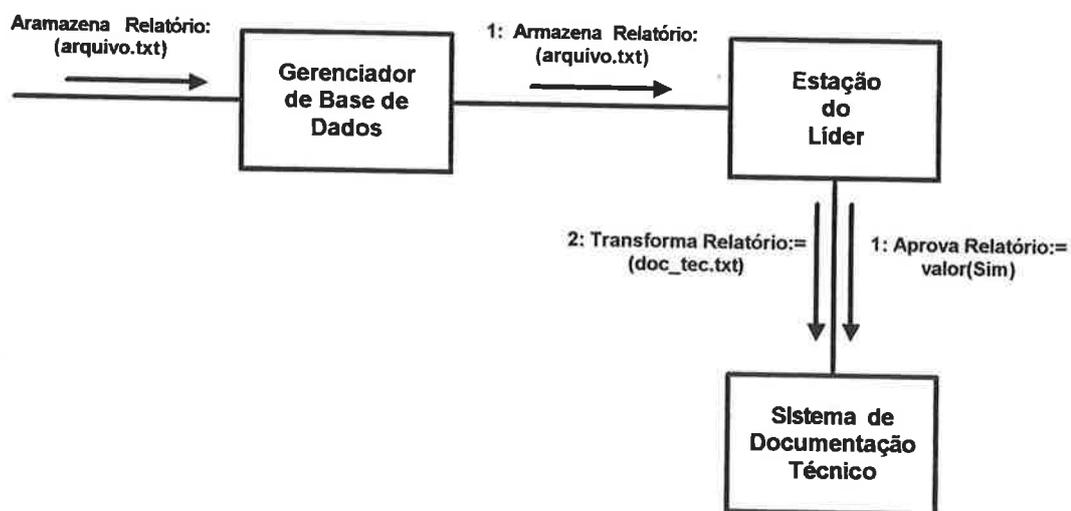


Figura 4.11 Diagrama de Colaboração do Esquema Estático

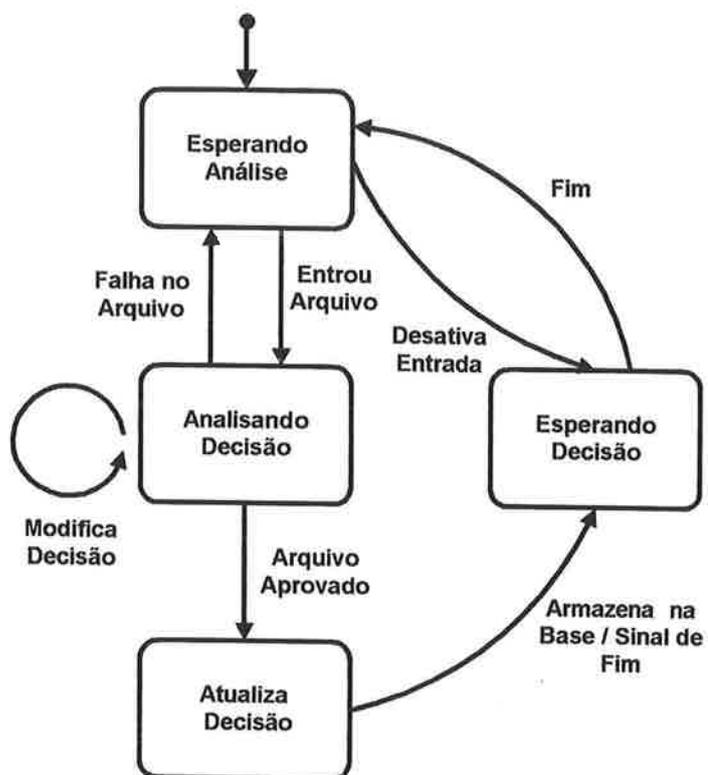


Figura 4.12 Diagrama de Estados de um Esquema Dinâmico

Finalmente, a figura 4.13 apresenta a definição formal de uma regra de informação, através de um Contrato, denominado de regra de análise de uma decisão de projeto.

O contrato está composto de: nome, responsabilidade, saída, exceção, pré-condição, e pós-condição. Estas informações deixam claro o entendimento da regra de informação.

4.4.3 Modelo da Computação

A figura 4.14 mostra o diagrama de objeto do ponto de vista da computação, este diagrama possui os seguintes elementos:

- Objetos computação - processador da equipe virtual: representam os processadores de informação distribuídos no ambiente de projeto;
- Objeto computação denominado de gerenciador da coordenação, encarrega-se do gerenciamento da conexão entre os objetos processador da equipe virtual;
- Interfaces: representam os tipos de informação trocadas entre objetos computacionais.

Este diagrama, define a distribuição computacional dos aplicativos de apoio a projeto, que devem implementar interfaces padronizadas segundo os tipos de informações manipuladas. A integração dos aplicativos é realizada através de um elemento denominado de gerenciador da coordenação

A figura 4.15 mostra o *Use Case* da regra da computação denominada de conexão primitiva, explicando os eventos que ocorrem no gerenciador de coordenação para executar uma conexão.

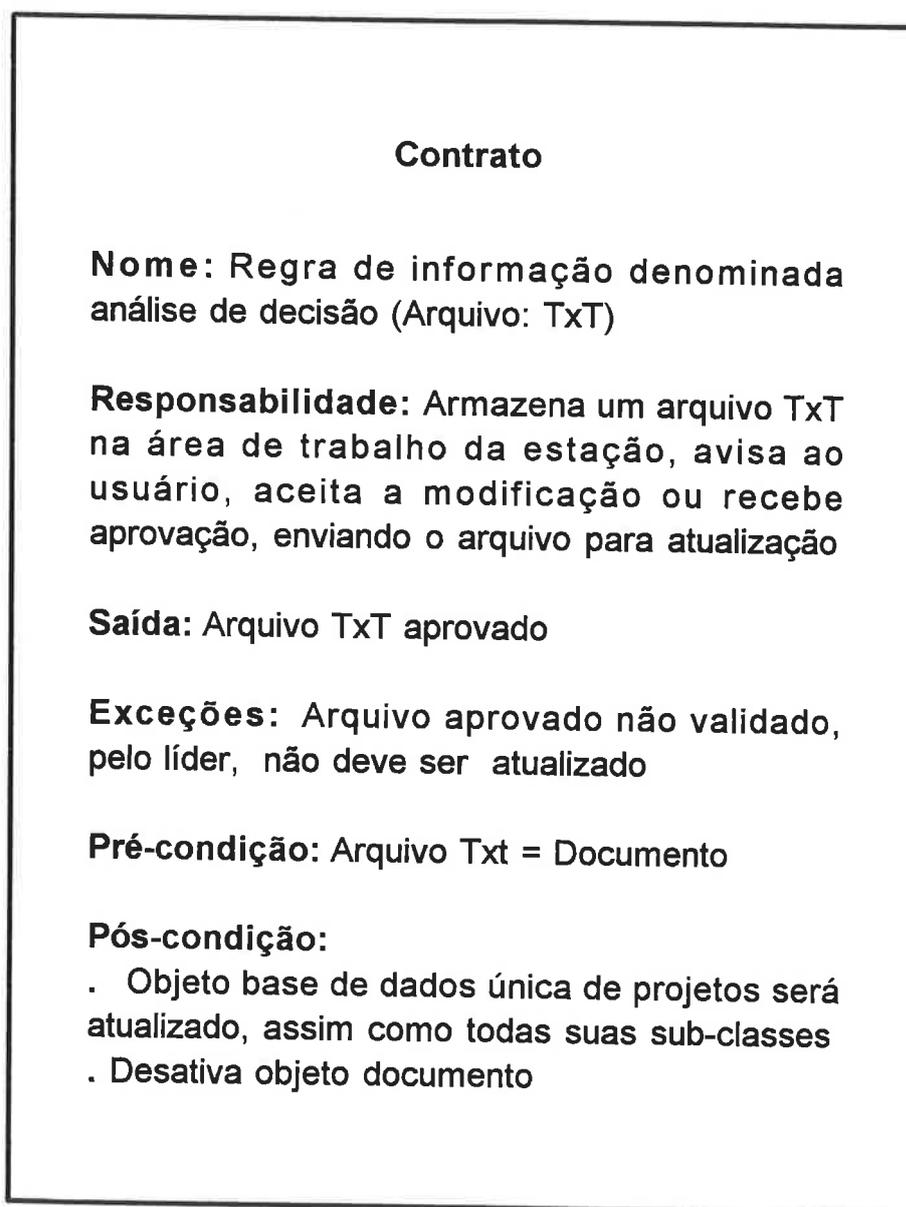


Figura 4.13 Contrato de uma Regra de Informação

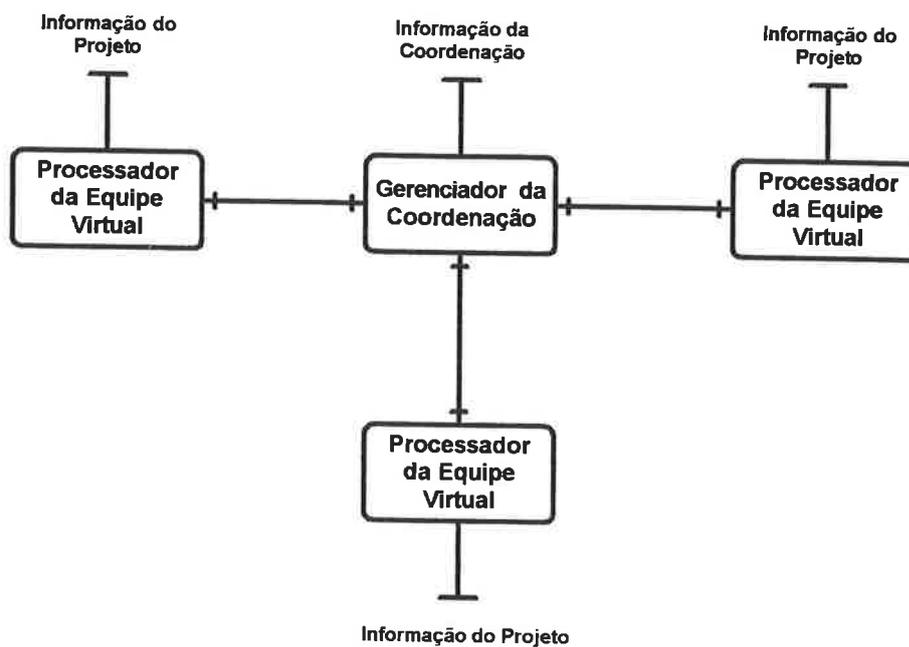


Figura 4.14 Diagrama de Objetos do Ponto de Vista Computacional

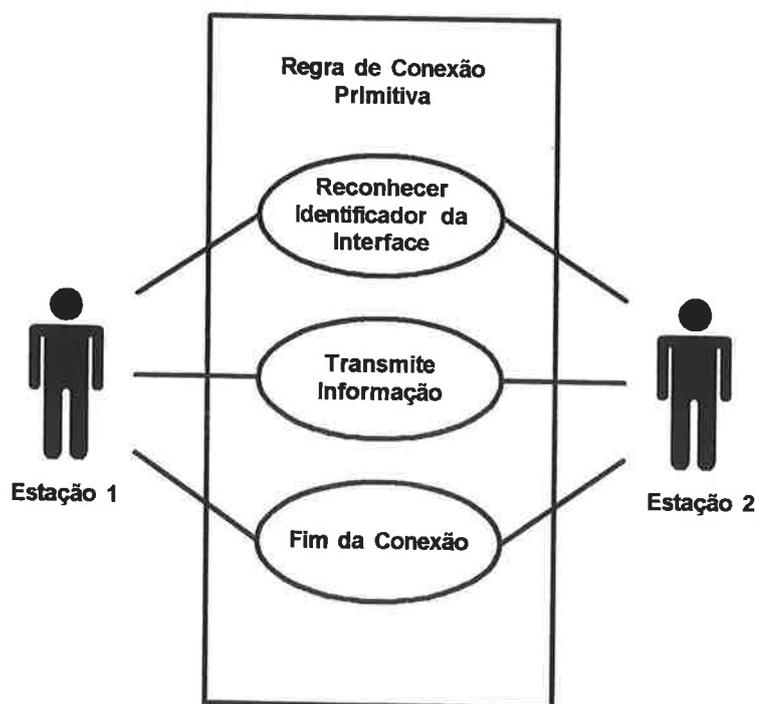


Figura 4.15 Use Case de uma Regra da Computação

A figura 4.16 mostra o contrato ou especificação da operação denominada, de reconhecimento do identificador de uma interface computacional, quando é realizada uma conexão.

As informações dos *Uses Cases* e dos Contratos completam as informações do modelo de objetos da computação aumentando o grau de completeza da especificação do ponto de vista da computação.

4.4.4 Modelo da Engenharia

Com relação ao ponto de vista da engenharia, o modelo está composto de um diagrama de objetos de engenharia, um diagrama *Use Case* e um contrato.

A figura 4.17 mostra o diagrama de objetos do ponto de vista da computação, que está constituído dos seguintes objetos engenharia:

- Objetos básicos de engenharia: constituído dos objetos computação, processador da equipe virtual e gerenciador da coordenação;
- Objetos do canal: objetos que são ativados para dar suporte à comunicação entre objetos básicos da engenharia. Estes objetos são o objeto adaptador, o objeto conector e o objeto protocolo, definidos no item 2.3.6.3;

Neste diagrama foram definidos dois tipos de canais de comunicação, o canal para fluxos de informação que indica a conexão entre o objeto produtor e o objeto consumidor é controlada pelo gerenciador da coordenação; e o canal de operação, que indica uma conexão direta entre o objeto cliente e o objeto servidor.

A análise do diagrama da figura 4.17, determina que os aplicativos distribuídos e o coordenador devem utilizar um esquema de comunicação baseados em

canais de comunicação dinâmicos, que se ativam quando existe uma transação cooperativa.

A figura 4.18 mostra o diagrama *Use Case* da regra de ativação e desativação do canal de comunicação. Nesta figura pode-se observar que na ativação de um canal pelo gerenciador do nó ou núcleo, devem ocorrer os seguintes eventos: um pedido de comunicação, a leitura da configuração das interfaces do objeto, ativação dos objetos do canal e o fim da comunicação.

Finalmente, a figura 4.19 mostra o contrato de uma regra de engenharia, denominada de migração de um objeto computação. Maiores informações sobre migração estão no item 2.4.2.

Nesta informação pode-se ressaltar a definição da pós-condição, que apresenta os estados que os objetos engenharia podem adotar após a execução da regra de engenharia.

4.4.5 Modelo da Tecnologia

O modelo do ponto de vista da tecnologia está constituído de um diagrama de objetos cujos elementos são os objetos tecnologia. A figura 4.20 mostra o diagrama de objetos do ponto de vista da tecnologia.

Estes objetos tecnologia, representam a tecnologia da informação, os quais implementam os objetos engenharia, por esta razão, o diagrama de objetos da tecnologia utiliza o diagrama de objetos da engenharia.

Neste diagrama, pode-se observar as possíveis tecnologias utilizadas no ambiente de projeto. Como exemplo, para o objeto processador da equipe virtual, utiliza-se um CASE específico para modelagem orientada a objetos, cuja plataforma será o Windows NT.

Contrato

Nome: Operação de reconhecimento de uma interface numa regra de conexão

Responsabilidade: Lê as mensagens, do objeto estação, e nos parâmetros determina o identificador da interface, imediatamente, compara com a informação que está na base de dados das interfaces e reconhece o tipo de identificador

Saída: Tipo de identificador

Exceções: Quando não existe o tipo de identificador da interface, deve encaminhar o reconhecimento para outro gerenciador de conexão

Pré-condição: Mensagem com os parâmetros: arquivo e identificador

Pós-condição:

- . A mensagem transforma-se num objeto informação de projeto
- . Ativa-se os objetos estações

Figura 4.16 Contrato de uma Regra de Computação

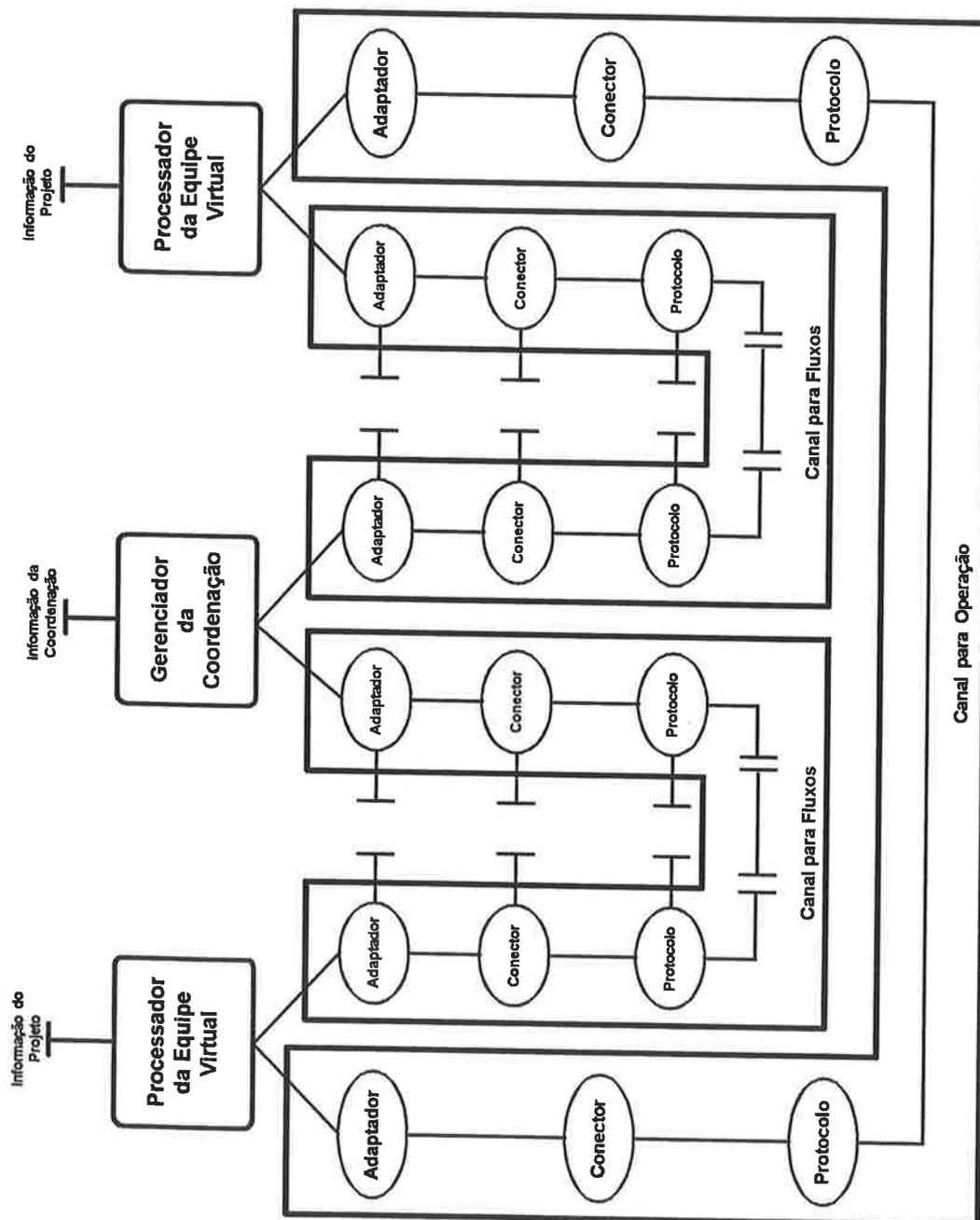


Figura 4.17 Diagrama de Objetos do Ponto de Vista da Engenharia

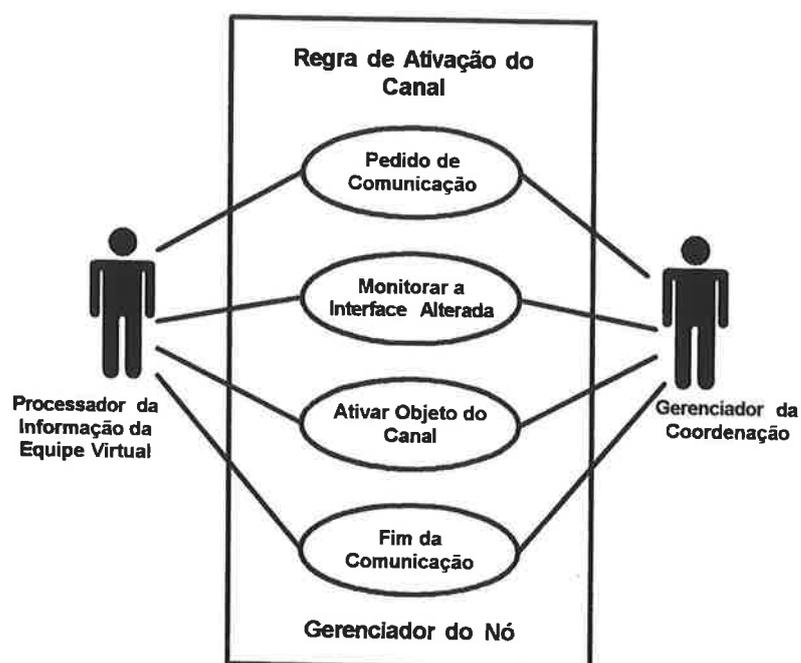


Figura 4.18 Use Case de uma Regra da Engenharia

Contrato

Nome: Reativação do canal por causa da migração de um objeto básico de engenharia

Responsabilidade: O objeto núcleo recebe uma notificação de migração, de parte de um objeto básico de engenharia, imediatamente desativa o canal atual, espera por um novo identificador e endereço

Saída: Autorização de conexão

Exceções: O objeto migrado não possui interface padronizada, gera uma mensagem de erro

Pré-condição: Informação de migração

Pós-condição:

- . Objetos dos canais são desativados
- . Objeto núcleo ativa a associação com o objeto básico de engenharia
- . Objeto dos canais são ativados

Figura 4.19 Contrato de uma Regra de Engenharia

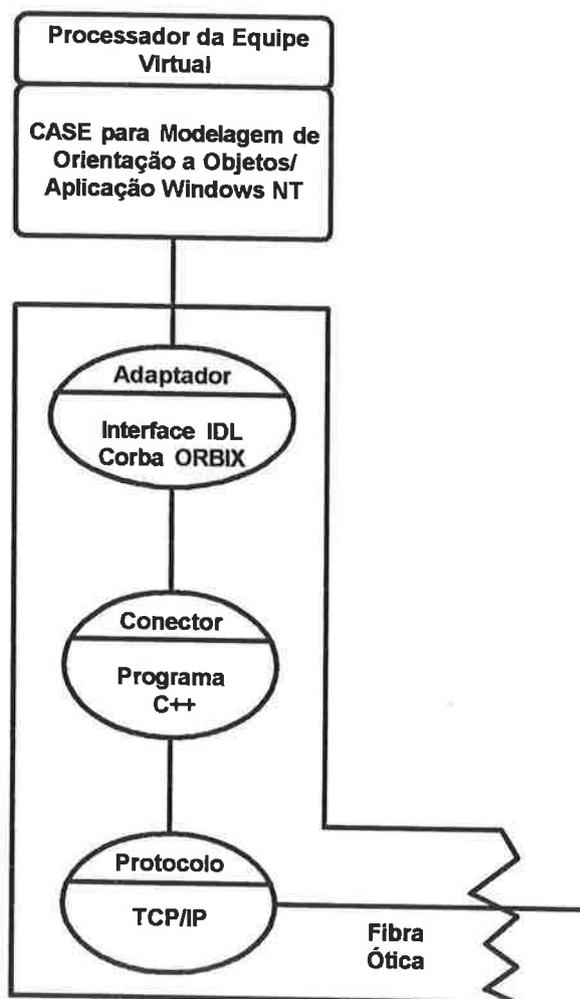


Figura 4.20 Diagrama de Objetos do Ponto de Vista da Tecnologia

Para o adaptador utiliza-se uma interface IDL (Interface Definition Language) da arquitetura CORBA da plataforma ORBIX, para o conector um programa C++ e para o protocolo utiliza-se os protocolos da arquitetura TCP-IP [Soares, 95], [Siegel, 96], [Gay, 95].

Os resultados da modelagem dos cinco pontos de vista mostram que a definição do sistema aberto de automação de projetos, tornou-se mais completa, pois, estas visões permitem ao projetista adquirir outros conhecimentos do sistema, que com as técnicas de especificação funcional [IEEE, 93] seria difícil de conseguir.

4.5 Arquitetura dos Níveis Hierárquicos

A definição da arquitetura do sistema aberto de automação de projetos utiliza o modelo apresentado no item 3.4 e o método descrito no item 3.5.

O resultado final desta atividade, será uma arquitetura de processadores, onde cada processador possuirá uma estrutura interna composta de objetos automação, os quais são derivados do modelo SAA e dos modelos dos cinco pontos de vista.

Cabe ressaltar que a análise dos níveis hierárquicos, atividade da fase denominada de projeto ODP, não será apresentada, pois estão totalmente discutidos e analisados nos trabalhos de [Matsuyama, 97] e [Tanomaru, 94].

Para encontrar a arquitetura de processadores, em cada nível hierárquico do SAA de projetos, realizou-se as seguintes tarefas [Kervella, 94a,b]:

- Identificou-se os objetos automação, os objetos computação e os objetos engenharia, que serão objetos aplicação ODP. Por exemplo, são objetos

aplicação ODP o objeto gerenciador de base de dados da coordenação e os objetos básicos de engenharia;

- Identificou-se os objetos computação e engenharia, que serão objetos gerenciador ODP. Neste caso, o objeto engenharia gerenciador da coordenação e os objetos do canal de comunicação correspondem ao objeto gerenciador ODP e constituem o objeto comunicação do SAA;
- Identificou-se os objetos informação como componentes do objeto automação gerenciador de base de dados.

Determinou-se a estrutura dos níveis hierárquicos do sistema aberto de automação, que é mostrada na figura 4.21. A estrutura básica, que está representada em todos os níveis, está constituída dos seguintes elementos:

- Objeto aplicação: representam os processadores de informação da equipe virtual de projeto. Cada nível possui seus objetos aplicação específicos;
- Objetos ODP: representam os aplicativos que implantam as funções ODP. A complexidade de implementação destas funções aumenta, segundo a hierarquia do nível, o nível de coordenação será o mais complexo;
- Objeto aplicação ODP: representa os aplicativos baseados no padrão ODP para aplicações de desenvolvimento de projeto;
- Objeto gerenciador ODP: representa o aplicativo gerenciador da interoperabilidade entre aplicativos distribuídos. Estes objetos estão distribuídos no canal permitindo a integração de todos os outros objetos. O desempenho destes objetos, deverá ser adequado, e dependerá da especificação dos serviços de coordenação do ambiente de projeto.

Finalmente, a figura 4.22 apresenta a arquitetura dos níveis hierárquicos do sistema aberto de automação de projeto, que está constituída de processadores interligados através de uma via de comunicação padrão [Nehmer, 92], [Robinson, 91]. Os elementos da arquitetura são definidos a seguir:

- Processador de gerenciamento do nível hierárquico: executa o processamento para as tarefas de coordenação, supervisão e controle das operações cooperativas do nível hierárquico respectivo;
- Processador de aplicação: executa as funções dos aplicativos que são utilizados pela equipe virtual no desenvolvimento do projeto. São os aplicativos computacionais de apoio ao planejamento, modelagem, especificação, entre outras atividades de apoio ao projeto;
- Processador ODP: executa o gerenciamento e operação das funções ODP, também executa os aplicativos padronizados pelo ODP, específicos da área de projeto;
- Processador gerenciador ODP: executa as funções referentes à interoperabilidade dos processadores distribuídos, implanta os protocolos de interoperabilidade entre processadores gerenciadores ODP utilizados na via de comunicação padrão [Rahkila, 97];
- Processador gerenciador de base de dados: executa o processamento referente ao gerenciamento da base de dados única de projeto;
- Processador de comunicação: executa os processamentos referentes aos protocolos de comunicação, utilizados na via de comunicação padrão, e implanta os protocolos específicos, de suporte aos serviços de comunicação cooperativa [Farooqui, 96], [Fatoohi, 97].

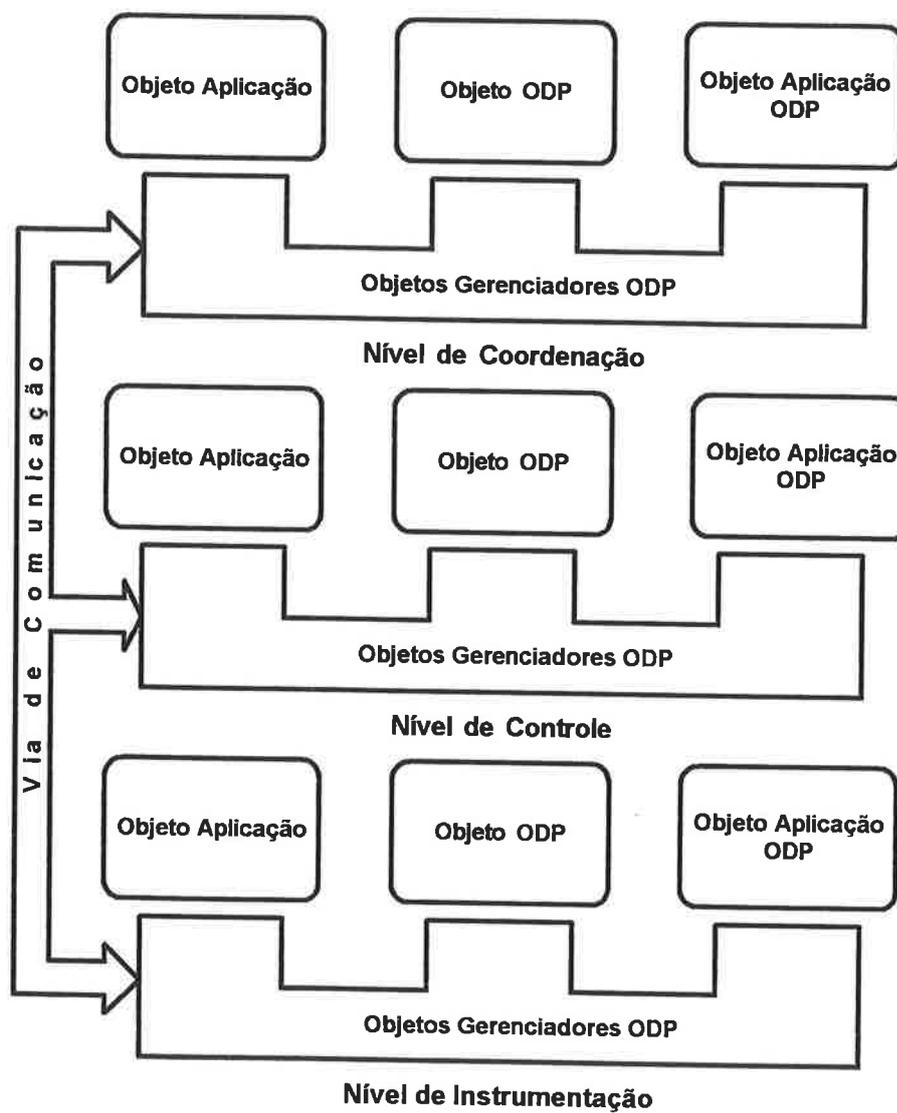


Figura 4.21 Estrutura dos Níveis Hierárquicos do Ambiente de Projeto

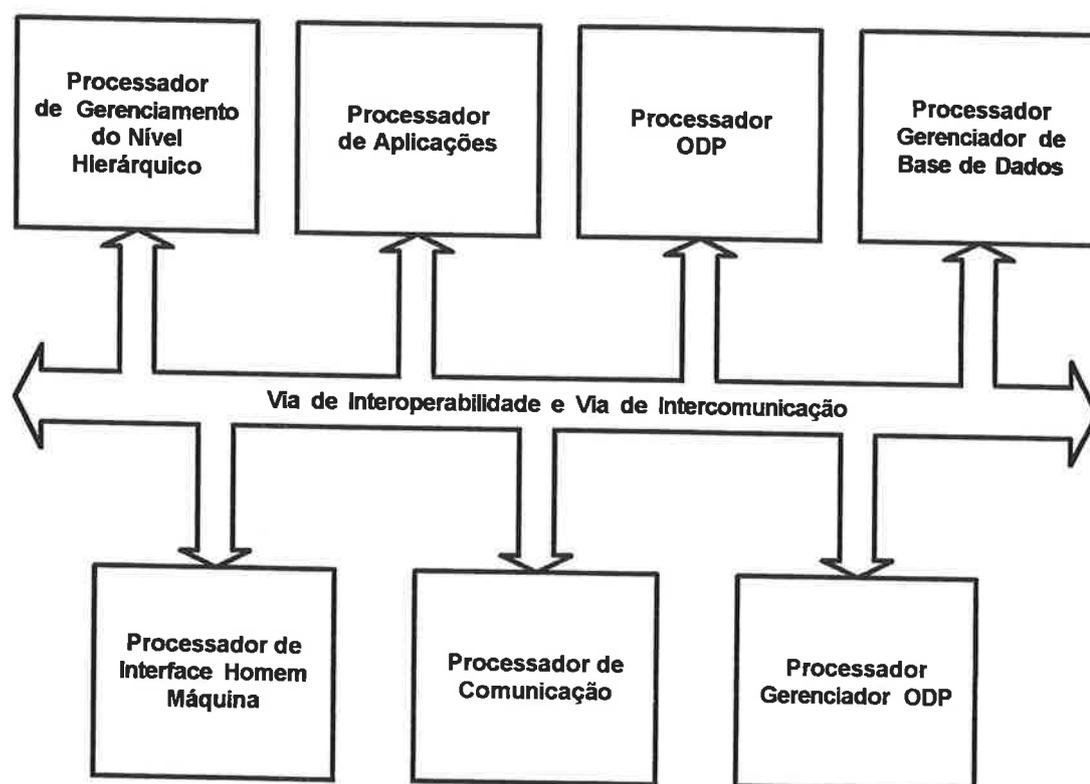


Figura 4.22 Arquitetura do SAA de Projeto

- Processador de interface homem-máquina: executa os processamentos das funções referentes às interfaces homem-máquina dos diferentes níveis hierárquicos;
- Via de comunicação: composta de duas vias, a via de interoperabilidade, cuja função principal é a integração dos aplicativos; e da via de intercomunicação, cuja função é o suporte da rede.

Esta estrutura básica é aplicada a cada nível hierárquico do sistema aberto de automação, mas sua implementação depende da especificação funcional de cada nível.

4.6 Especificação da Implementação

Denomina-se especificação da implementação ao nível de abstração do sistema que corresponde à seleção das tecnologias de informação que implementam os elementos da arquitetura do sistema.

Este item é fortemente relacionado com a especificação do ponto de vista da tecnologia, basicamente este ponto de vista representa a especificação prioritária sobre os outros pontos de vista, como foi definido na tabela 2.1.

No que diz respeito ao conceito de aberto, as soluções tecnológicas aqui propostas utilizam padrões internacionais ou em processo de padronização, mas em alguns casos, pode-se recomendar a utilização de soluções proprietárias.

A figura 4.23 apresenta, a estrutura do sistema aberto de automação do ambiente de projetos, cujos elementos são:

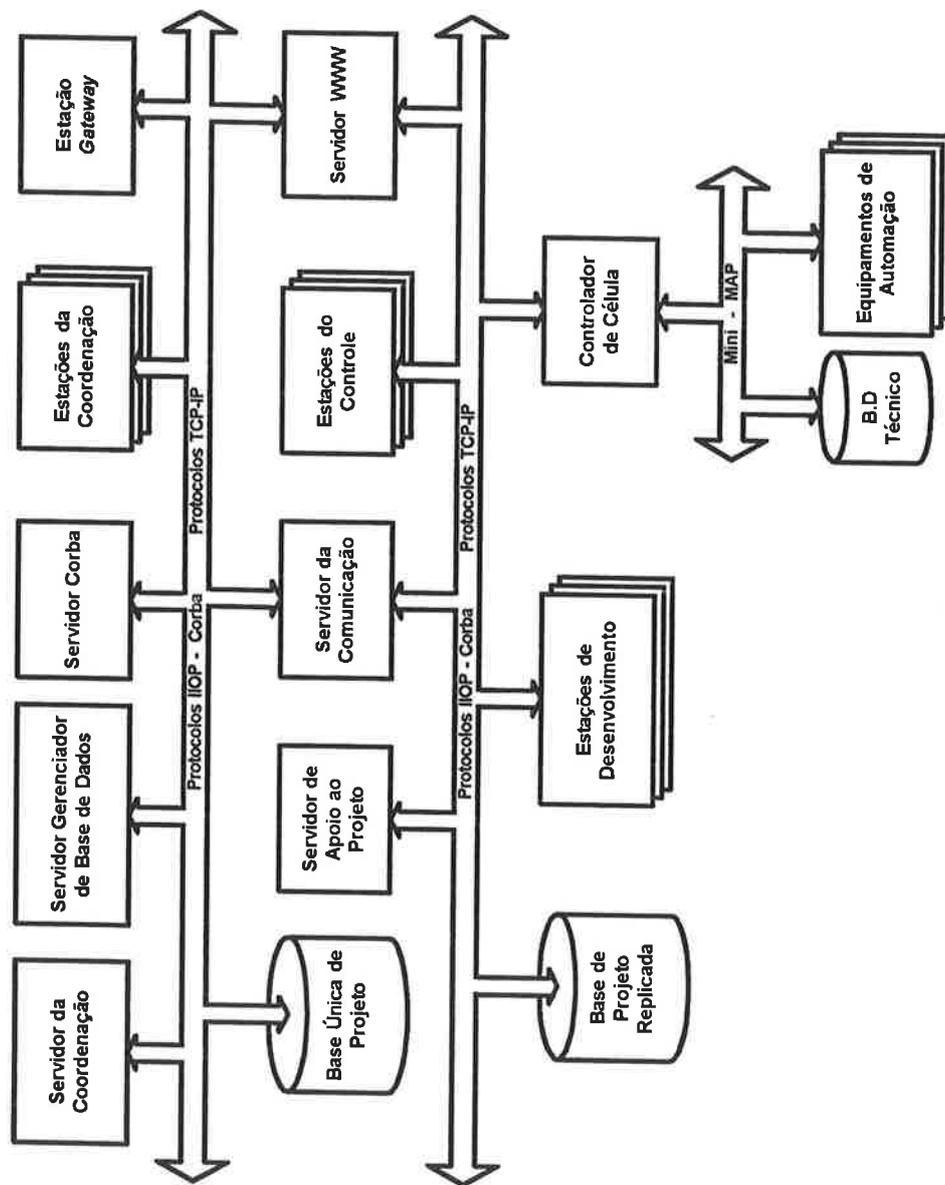


Figura 4.23 Estrutura do Sistema Aberto de Automação de Projetos

- Servidor da coordenação: os servidores definidos nesta estrutura estão compostos de sub-sistemas (módulos processadores), interligados por vias de comunicação padrão, da forma como foi proposta em [Martucci, 92];

Este servidor implanta as funções de gerenciamento do nível de coordenação e da operação central do sistema corporativo;

- Servidor gerenciador da base de dados: implementa os modelos relacional e orientados a objetos como estruturas padrão, implanta as funções de gerenciamento de base de dados e utiliza o SQL (Structure Query Language) como linguagem padrão.

Para a distribuição otimizada da informação, utiliza-se uma arquitetura para manipular grandes quantidades e complexidades de dados (Data Warehouse) e permite a criação de bases de dados personalizadas, a partir da base única de projeto. Estas bases são distribuídas no ambiente cooperativo [Gelman, 96].

Com relação aos formatos de dados e documentos manipulados, utiliza-se os padrões ODA (Open Architecture Document), EDI (Electronic Data Interchange) e PDES (Product Data Electronic Exchange), [Appelt, 93], [Baier, 95], [Hunter, 89], [Davis, 91];

- Servidor CORBA: implanta em sua estrutura o ambiente Orbix que implementa o padrão CORBA 2.0. É a plataforma básica da arquitetura CORBA, gerência as interfaces IDL (Interface Definition Language), gerência o protocolo de interoperabilidade IIOP (Internet InterORB Protocol), e implanta as funções ODP [Eckert, 96];
- Servidor de comunicação: esta estrutura implanta as aplicações da arquitetura TCP-IP, como correio eletrônico, FTP (File Transfer Protocol),

- mecanismos de segurança, e, principalmente, gerência o serviço de vídeo conferência, neste serviço, é utilizado o protocolo internacional H.323 da ITU [Lins, 97], [Diaz, 93], [Gale,92];
- Servidor WWW (World Wide Web): implementa as funções referentes aos serviços da tecnologia Web (OrbixWeb), implanta os protocolos HTTP (Hyper Text Transfer Protocol) e disponibiliza as informações de projeto manipuladas pela equipe virtual [Kiname, 96], [Leydekkers, 94];
 - Estações da coordenação: são plataformas onde são implantados os aplicativos distribuídos de gerenciamento de cada departamento da corporação e outros aplicativos cooperativos, como pode-se ver em [Dewan, 93], [Chandrasekaran, 93]. Estas plataformas implantam as interfaces IDL da arquitetura CORBA 2.0;
 - Estação *Gateway*: esta plataforma permite a interligação do sistema de automação com outros sistemas de automação, não necessariamente baseados na arquitetura CORBA.
- Suporta os programas de integração para gerenciamento, baseados nos protocolos SNMP (Simple Network Management Protocol) e CMIP (Common Management Information Protocol), dos grupos JIDM (Join Domain Management - CORBA) e IIMC (ISO an Internet Management Coexist) [Narasimhan, 97], [Berson, 96], [Barr, 93];
- Estações do Controle: são plataformas CORBA que implantam as funções de coordenação setorial e da supervisão local;
 - Servidor de apoio ao projeto: implementa ferramentas computacionais de apoio ao projeto;

- Estações de desenvolvimento: são plataformas CORBA utilizadas pela equipe de desenvolvimento.

A via de comunicação que interliga todos os elementos da equipe virtual, utiliza os protocolos de comunicação da arquitetura TCP-IP, o protocolo de interoperabilidade IOP da arquitetura CORBA, e utiliza redes físicas de alta velocidade como o FDDI ou DQDB [Heffmann, 90], [Heinrichs, 93], [Shiratori, 96].

Com relação a estação da produção e aos instrumentos da sub-rede da produção, que é um ambiente voltado a produção, utiliza-se uma plataforma baseada no padrão IEEE P-1014, com sistema operacional específico ao produto ou serviço a produzir, suportando protocolos de comunicação padrão TCP-IP, MAP, Mini-Map e Fieldbus, como foi analisado em [Becerra 93].

Dada a existência de soluções tecnológicas para implementar a arquitetura do ambiente de projeto, conclui-se que a implementação de um ambiente de projeto aplicado aos grandes empreendimentos é tecnologicamente viável.

Com esta fase, finaliza-se a experimentação da metodologia baseada no padrão ODP. A seguir, são discutidos os principais resultados deste processo.

4.7 Análise dos Resultados

O experimento da metodologia ODP gerou uma série de modelos, diagramas e informações de projeto, que se tornam subsídios para dissertar sobre a aplicabilidade do padrão ODP nos projetos de automação.

A aplicabilidade do padrão ODP será medida indiretamente, através da aplicação da metodologia ODP e dentro deste contexto restrito, serão definidos variáveis qualitativas e heurísticas.

Estas variáveis heurísticas, denominadas de variáveis de aplicabilidade, serão os pontos de comparação com outras formas de desenvolvimentos de projetos de automação, apresentadas em [Tanomaru, 94], [Matsuyama, 97] e [Becerra, 93].

A identificação das variáveis de aplicabilidade são consequência da análise feita em outros tipos de projetos e baseada na experiência real do autor em diversos projetos.

As variáveis de aplicabilidade são organizadas em grupos, que são definidos segundo os seguintes aspectos de projeto: especificação, níveis de abstração, requisitos do usuário, modelos, implementação, ferramentas, domínio de aplicação, gerenciamento, e qualidade. A seguir são definidas todas as variáveis:

- Clareza (especificação): refere-se à forma clara e precisa de apresentar as informações da especificação do sistema;
- Completeza (especificação): refere-se à quantidade de informação necessária para poder entender o sistema;
- Identificação dos requisitos (requisitos): refere-se às facilidades oferecidas para poder detectar os requisitos dos usuários;
- Aquisição do conhecimento (requisitos): refere-se às facilidades para poder adquirir o conhecimento, referente ao sistema, no contexto da aplicação;
- Controle dos requisitos (requisitos): refere-se à capacidade de suportar mecanismos de controle dos requisitos;
- Evolução uniforme (nível de abstração): refere-se à forma seqüencial e uniforme, como os níveis de abstração do sistema evoluem;

- Homogeneização (nível de abstração): refere-se as semelhanças das linguagens abstratas utilizadas em cada nível de abstração do sistema;
- Complexidade (modelos): refere-se à complexidade do entendimento dos modelos utilizados;
- Definição da arquitetura (modelos): refere-se aos mecanismos de definição da arquitetura do sistema;
- Estruturação (implantação): refere-se às facilidades para poder especificar e planejar a implementação da arquitetura do sistema;
- Automatização (Ferramentas): refere-se às facilidades para incluir ferramentas computacionais de apoio aos processos;
- Abrangência (domínio de aplicação): refere-se à capacidade de poder utilizar e adaptar a metodologia em outras aplicações;
- Organização da informação (gerenciamento): refere-se às facilidades para poder organizar as informações do desenvolvimento;
- Complexidade dos documentos (gerenciamento): refere-se à complexidade do conteúdo das informações de projeto;
- Pontos de controle (gerenciamento): refere-se às facilidades para poder configurar pontos de controle ao longo do desenvolvimento;
- Padronização (qualidade): refere-se a adoção de padrões nas atividade das fases da metodologia;

- Testes (qualidade): refere-se à complexidade dos mecanismos de testes utilizados em cada fase da metodologia;

A quantificação das variáveis de aplicabilidade depende da realização de outros experimentos, mas adotou-se os seguintes tipos de unidades: baixo, médio e alto. Estes tipos de unidades representam parâmetros de referência para analisar a aplicabilidade da metodologia.

A tabela 4, 1 apresenta os resultados das variáveis de aplicabilidade, em três tipos de projeto: projeto ODP, projeto não ODP, projeto estruturado. Estes resultados são produtos de experiências reais e acadêmicas.

Nesta tabela, pode-se observar que o projeto tipo ODP possui melhores condições que os outros tipos de projeto.

Com relação a especificação do sistema, a utilização do modelo SAA e dos pontos de vista, aumentaram a capacidade de definição e controle dos requisitos do sistema, aumentando a garantia da qualidade. Este é o ponto forte desta metodologia.

No projeto ODP existe um desenvolvimento homogêneo do ciclo de vida, existem fases corretamente seqüenciais e além disso, utiliza-se modelos orientados a objetos desde o início até o fim do projeto, característica que o diferenciam dos outros tipos de projeto.

Sobre a arquitetura do sistema, esta metodologia ODP é a única que apresenta uma estratégia formal, pois permite o mapeamento da análise do sistema numa estrutura arquitetônica padrão, como é a arquitetura CORBA.

Variáveis	Tipos de Projeto	Projeto ODP	Projeto não ODP	Projeto Estruturado
Clareza		Alto	Médio	Baixo
Completeza		Alto	Médio	Baixo
Identificação dos Requisitos		Alto	Médio	Médio
Aquisição dos Conhecimentos		Alto	Baixo	Médio
Controle dos Requisitos		Médio	Médio	Baixo
Evolução Uniforme		Alto	Alto	Baixo
Homogeneização		Alto	Médio	Baixo
Complexidade dos Modelos		Alto	Médio	Médio
Definição da Arquitetura		Alto	Baixo	Baixo
Estruturação da Implementação		Alto	Alto	Alto
Automatização		Alto	Alto	Alto
Abrangência		Médio	Médio	Alto
Organização da Informação		Médio	Médio	Médio
Complexidade dos Documentos		Alto	Médio	Baixo
Pontos de Controle		Alto	Alto	Alto
Padronização		Alto	Alto	Médio
Testes		Baixo	Baixo	Baixo

Tabela 4.1 Resultados do Experimento ODP

Com relação à gerência de projeto, este tipo de projeto possui todas as características dos outros tipos de projeto, mas exige do gerente de projeto conhecimentos profundos sobre os fundamentos ODP.

Dentro deste contexto, o padrão ODP torna-se uma ferramenta poderosa no desenvolvimento de projetos de sistemas distribuídos, mas deve ser utilizado dentro de um esquema formal (metodologia) para poder obter esses benefícios comentados anteriormente.

Cabe ressaltar que parte dos resultados apresentados foram confirmados com aplicação desta metodologia ODP em dois projetos reais. Um projeto na área da automação comercial, um sistema de comércio eletrônico baseado em tecnologias WWW e Internet; e outro, na área de telecomunicações, um sistema de integração de sistemas TMN (Telecommunication Management Network), baseados na arquitetura CORBA.

Este capítulo apresentou o experimento da metodologia ODP e a proposta de sistema aberto de automação aplicado a um ambiente de projetos que suporta a engenharia simultânea. A seguir, serão apresentadas as conclusões e comentários finais deste trabalho de tese.

Capítulo 5 Considerações Finais

Neste capítulo final apresenta-se as principais conclusões encontradas neste trabalho de pesquisa, assim como, serão apresentados os comentários gerais, uma estrutura de pesquisa gerada para impulsionar próximos trabalhos e a contribuição acadêmica.

5.1 Conclusões

As conclusões da tese são definidas nos seguintes tópicos: a aplicabilidade do padrão ODP nos projetos de sistemas de automação, a metodologia ODP e a complexidade dos requisitos do sistemas, os pontos de vista e seus benefícios, e a solução de automação do ambiente com engenharia simultânea.

A aplicação do padrão ODP nos projetos de sistemas de automação, da forma como foi apresentada na norma [ISO, 95c], não seria uma atividade simples e rápida, apesar do forte embasamento conceitual que esta norma possui, encontrou-se que a aplicabilidade, depende necessariamente da estruturação de uma metodologia de desenvolvimento adequada, onde será inserida a proposta ODP.

Foi demonstrado que a inclusão do padrão ODP potencializa a efetividade da metodologia de projeto, que por sua vez se transforma na ferramenta que possibilita a aplicabilidade do padrão. Atualmente, na mesma linha de atuação, encontra-se o trabalho de aplicação do ODP na área de telecomunicações, realizado pelo grupo 15 da ITU [ITU, 96].

Uma conclusão importante está na relação da metodologia ODP com os requisitos do sistema. Um sistema distribuído e aberto engloba muitos aspectos que tornam difícil a definição dos requisitos, na metodologia ODP os pontos de vista, o modelo de referência SAA e arquitetura CORBA, tornam esta tarefa

muito simples e completa, permitindo a manipulação adequada dos requisitos do sistema, desde a fase de especificação até a implementação do sistema.

Um tópico essencial na conclusão desta tese são os pontos de vista, estes são o centro da proposta ODP, pois através de sua utilização foi possível definir de forma completa o sistema de automação e permitiu gerar informações, que se tornaram subsídios iniciais, nas outras fases do desenvolvimento.

Mas de qualquer forma, para que os pontos de vista sejam verdadeiras peças chaves nos projetos de sistemas ODP deverão ser utilizados de forma estruturada e sistemática.

Cabe ressaltar, que o ponto de vista da empresa transformou-se num aspecto crítico, de vital importância para os outros pontos de vista. Este ponto da empresa, definiu o contexto principal que engloba os outros pontos de vista, e, principalmente, controla a sintonia do projeto com a corporação.

Dentro do contexto da solução do ambiente de projeto, que suporta engenharia simultânea, encontrou-se que este tipo de ambiente representa uma aplicação dos sistemas cooperativos e que sua implementação tem que se realizar, dentro do escopo do padrão ODP e dos sistemas abertos, fora desse escopo representa uma tarefa complicada, e, dependendo do tamanho da corporação, inviável.

Finalmente, o sucesso da implantação da engenharia simultânea depende basicamente do sistema de automação que o suporta, é uma relação muito forte, portanto, o projeto do sistema de automação deve ser realizado metodologicamente.

5.2 Comentários Gerais

Um comentário que merece ter destaque é a linha acadêmica que este trabalho de pesquisa adotou.

Esta linha inicia-se numa tese de livre docência e durante sua evolução, gera trabalhos de pesquisa em todos os níveis, mestrado, doutorado e iniciação científica, onde o autor atuou direta ou indiretamente, e chega neste ponto, com este trabalho de tese, que utiliza os subsídios e demonstrações anteriores, colocando a linha acadêmica num ponto de crescimento natural, que pode ser chamado de novo impulso acadêmico.

O descrito anteriormente mostra o ciclo de desenvolvimento da pesquisa, cujos produtos gerados (teses, projetos, recursos humanos, entre outros), fortalecem o papel da universidade dentro da comunidade nacional.

Outro comentário, que merece ser observado, está nos conceitos sobre orientação a objetos e engenharia de software. Este trabalho de tese utiliza a orientação a objetos como uma ferramenta, impulsionando a aplicação deste paradigma em sistemas distribuídos e abertos.

Os conceitos da engenharia de software foram aplicados de forma adequada para uma visão de sistema, ressaltando a importância da relação do desenvolvimento de projetos de sistemas com esta disciplina de software.

É importante ressaltar, que ao longo do trabalho, foram incluídas novas tecnologias, um desses destaques é a arquitetura CORBA, atualmente em evolução, que gerou todo um estudo sobre sua relação com a proposta ODP. Esta arquitetura terminou por inspirar o modelo arquitetônico da tese.

Outro destaque foi a inclusão de tecnologias Internet, no início do desenvolvimento da tese, existiam aplicações muito restritas, atualmente se transformam em elementos básicos do ambiente de projeto.

Estes seriam os pontos mais importantes, que o autor queria mostrar sobre o trabalho de tese, que poderão servir como referência para outros trabalhos de pesquisa.

5.3 Continuidade da Pesquisa

Este trabalho de tese inicia um novo impulso na linha de pesquisa de sistemas abertos, essa continuidade, representa esforços em todos os níveis acadêmicos, doutorado, mestrado e iniciação científica, assim como, esforços conjuntos com a comunidade industrial.

A figura 5.1 apresenta a estrutura de pesquisa, no topo está a linha principal, denominada de sistema distribuído e aberto ODP, que envolve os tópicos relacionados com esta tese, enfatizando-se as aplicações em sistemas de automação.

Em continuidade são identificados quatro áreas de trabalho: o padrão ODP, a orientação a objetos distribuídos, a engenharia simultânea e a arquitetura CORBA.

Com relação ao padrão ODP dar-se-á continuidade com a pesquisa dos inter-relacionamento dos cinco pontos de vista e sua aplicação na engenharia dos requisitos de sistemas.

Estes objetivos serão atingidos com aplicações específicas de cada ponto de vista, em diferentes domínios de aplicação, estas atividades são representadas nas caixas da figura 5.1.

Na área de orientação a objetos aumentar-se-á as pesquisas com o objetivo de equacionar os benefícios da orientação a objetos aplicados a sistemas

distribuídos. Estes objetivos serão atingidos realizando projetos distribuídos baseados em OO.

O interesse da engenharia simultânea está na implementação real de ambientes de projetos, procurando aprofundar no *Workflow* da coordenação. Aplicações diversas permitirão atingir estes objetivos.

Com relação, a arquitetura CORBA, atualmente centro de pesquisas, visa-se orientar sua aplicação a diferentes sistemas de automação e contribuir na sua consagração.

Finalmente, todos estes esforços acadêmicos e técnicos devem convergir para um objetivo final, representado na base da figura, e denominado de, estruturação de um sistema aberto, cooperativo e corporativo (GroupWare), que poder-se-á configurar para diferentes tipos de domínio de aplicação.

5.4 Contribuição Acadêmica

A contribuição acadêmica define a forma como este trabalho de tese aumentou as fronteiras do conhecimento, tornando-a inédita dentro da comunidade científica.

O ponto mais importante da contribuição acadêmica, está na aplicabilidade do padrão ODP, esta proposta deverá contribuir na sua consagração e na sua aceitação.

Outra contribuição está na arquitetura do ambiente de projeto que suporta engenharia simultânea, esta tese representa uma proposta que gerará outras propostas práticas na comunidade industrial.

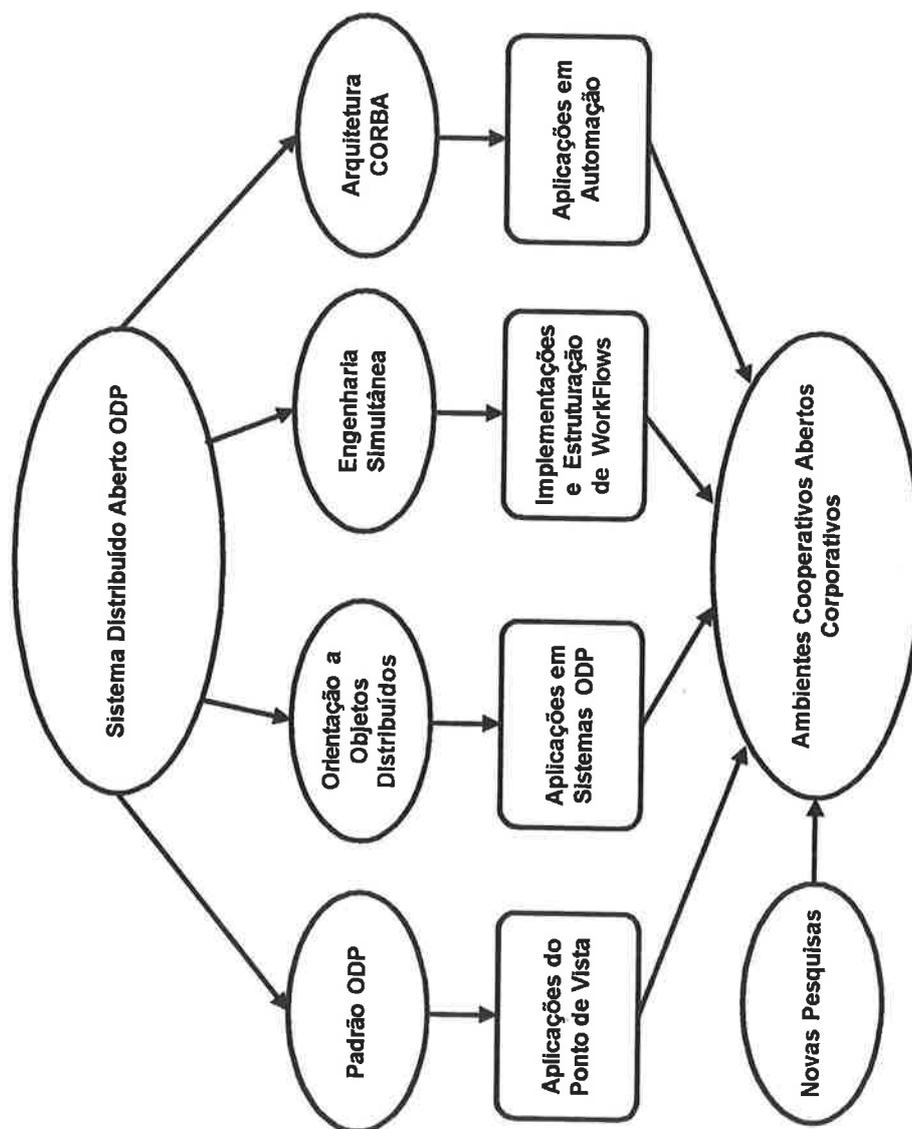


Figura 5.1 Estrutura de Pesquisa

Existe ainda muitos aspectos para pesquisar na arquitetura CORBA, esta tese contribui neste sentido, apresentando conclusões na aplicação em sistemas de automação.

Em orientação a objetos a contribuição está no aprimoramento da forma de aplicação, formalizando-se o uso em projetos de sistemas de automação.

Finalmente, esta tese deve contribuir e impulsionar as atividades de graduação e pós-graduação do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo.

Referências Bibliográficas

- [Adler, 95] Adler, R. Distributed coordination models for cliente/server computing. **Computer**, v. 28, n. 4, p. 14-22, 1995.
- [Appelt, 93] Appelt, W.; Tetteh-Lartey, N. The formal specification of the ISO open document architecture (ODA) standard. **The Computer Journal**, v. 36, n. 3, p. 269-79, 1993.
- [Awad, 96] Awad, M.; Kuusela, J.; Ziegler, J. **Object-oriented technology for real-time systems: a practical approach using OMT and Fusion**. 1.ed. New Jersey, Prentice Hall PTR, 1996.
- [Baier, 95] Baier, R.; Moeller, E.; Scheller, A. ODA implementation for multimedia teleservices. **Computer Communications**, v. 18, n. 12, p. 889-95, 1995.
- [Bailetti, 94] Bailetti, A.J.; Callahan, J.R.; Dipietro, P. A coordination structure approach to the management of projects. **IEEE Transactions on engineering management**, v. 41, n. 4, p. 394-403, 1994.
- [Barr, 93] Barr, W.J.; Boyd, T.; Inoue, Y. The TINA initiative. **IEEE Communications Magazine**, p. 70-6, 1993.
- [Becerra, 93] Becerra, J.L.R. **Especificação, projeto e implementação de um controlador de célula flexível aplicado na manufatura**. São Paulo, 1993. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo.
- [Becerra, 94] Becerra, J.L.R.; Martucci Jr., M. Implementação de Arquiteturas Abertas: uma Aplicação na Automação da Manufatura. **Boletim da Escola Politécnica**, 1994. /Separata/
- [Becerra, 96 a] Becerra, J.L.R.; Martucci Jr., M. Los sistemas abiertos en la automatización: una experiencia aplicada en la manufatura. **Revista Mexicana de Ingeniería Electrónica**, v. 1, n. 1, p. 1-4, 1996.
- [Becerra, 96 b] Becerra, J.L.R.; Martucci Jr., M. Modelagem de um sistema aberto de automação que suporta engenharia simultânea. In: CONGRESSO LATINOAMERICANO DE CONTROL AUTOMÁTICO, 7., Buenos Aires, 1996. **Anais**.

- [Becerra, 97] Becerra, J.L.R.; Martucci Jr., M. Propuesta de arquitectura abierta para sistemas de automatización. In: SIMPOSIO INTERNACIONAL DE AUTOMATIZACION, 2., Lima, 1997. **Anais**. p. B-1.1 - B-1.6.
- [Benford, 92] Benford, S.; Smith, H.; Shepherd, A.; Bullock, A.; Howidy, H. Information sharing approach to CSCW: the grace project. **Computer Communications**, v. 15, n. 8, p. 503-08, 1992.
- [Bentley, 94] Bentley, R.; Rodden, T.; Sawyer, P.; Sommerville, I. Architectural support for cooperative multiuser interfaces. **Computer**, v. 27, n. 5, p. 37-46, 1994.
- [Bernstein, 96] Bernstein, L. Software in the large. **AT&T Technical Journal**. p. 5-14, 1996.
- [Berra, 90] Berra, P.B.; Chen, C.Y.R.; Ghafoor, A.; Lin, C.C.; Little, T.D.C.; Shin, D. Architecture for distributed multimedia database systems. **Computer Communications**, v. 13, n. 4, p. 217-30, 1990.
- [Berson, 96] Berson, A. **Client/server architecture**. 2.ed. New York, McGraw-Hill, 1996.
- [Blair, 94] Blair, G.; Dark, P.; Davies, N.; Mariani, J.; Snape, C. Integrated support for complex objects in a distributed multimedia design environment. **Computer Science**, p. 30-7, 1994.
- [Bond, 97] Bond, A.; Arnold, D.; Chilvers, M. Designing and building an ODP environment. [online]. p. 1-13. Available: <http://www.dstc.edu.au/Hector/papers/icodp97/icodp97.html>, february 1998.
- [Booch, 94] Booch, G. **Object-oriented analysis and design: with applications**. 2.ed. California, The Benjamin/Cummings, 1994.
- [Bowen, 91] Bowen, D. Open distributed processing, **Computer Networks and ISDN Systems**, v. 23, p. 195-201, 1991.
- [Carlson, 96] Carlson, A.C.; Brook, W.R.; Haynes, C.L.F. Experiences with distributed objects. **AT&T Technical Journal**, p. 58-67, 1996.

- [CERC, 93] CONCURRENT ENGINEERING RESEARCH IN REVIEW. **Commercial off-the-shelf software for concurrent engineering**. v. 5, p. 15-21, 1993. (CERC Report for Task 3.1. Phase 5 of the DARPA Initiative in Concurrent Engineering, FY 1993)
- [Chandrasekaran, 93] Chandrasekaran, B.; Goel, A.K.; Iwasaki, Y. Functional representation as design rationale. **Computer**, v. 26, n. 1, p. 48-56, 1993.
- [Chang, 93] Chang, A.; Bailey, A.D.; Whinston, A.B. A distributed knowledge-based approach for planning and controlling projects. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 23, n. 6, p. 1537-50, 1993.
- [Coad, 91] Coad, P.; Yourdon, E. **Object-oriented analysis**. 1.ed. New Jersey, Prentice-Hall, 1991.
- [Costa, 94] Costa, C.P. Engenharia simultânea e workgroup computing. In: CONGRESSO INTERNACIONAL DE COMPUTAÇÃO GRÁFICA. **Anais**. São Paulo, 1994.
- [Coulson, 95] Coulson, G.; Blair, G.S.; Stefani, J.B.; Horn, F.; Hazard, L. Supporting the real-time requirements of continuous media in open distributed processing. **Computer Networks and ISDN Systems**, v. 27, p. 1231-46, 1995.
- [Crowcroft, 95] Crowcroft, J. **Open distributed systems**. 1.ed. London, UCL Press, 1995.
- [Cutkosky, 93] Cutkosky, M.R.; Engelmores, R.S.; Fikes, R.E.; Genesereth, M.R.; Gruber, T.R.; Mark, W.S.; Tenenbaum, J.M.; Weber, J.C. PACT: An experiment in integrating concurrent engineering systems. **Computer**, v. 26, n. 1, p. 28-37, 1993.
- [Davis, 91] Davis, T.; Trapp, G. Advancing concurrent engineering using STEP. Morgantown, **Concurrent Engineering Research in Review**, 1991. (Technical Report RN-91-012)
- [Davis, 93] Davis, A.M. **Software requirements: objects, functions, and states**. 1.ed. New Jersey, Prentice Hall PTR, 1993.
- [Dewan, 93] Dewan, P.; Riedl, J. Toward computer-supported concurrent software engineering. **Computer**, v. 26, n. 1, p. 17-27, 1993.

- [Diaz, 93] Diaz, M.; Villemur, T. Membership services and protocols for cooperative frameworks of processes. **Computer Communications**, v. 16, n. 6, p. 548-56, 1993.
- [Dwivedi, 91] Dwivedi, S.N.; Sobolewski, M. Concurrent engineering – an introduction. In: INTERNATIONAL CONFERENCE ON CAD/CAM ROBOTICS AND FACTORIES OF THE FUTURE, 5., New York, 1991. **Proceedings**. Springer-Verlag, 1991. v. 1, 16 pp.
- [Eckert, 96] Eckert, K.P. From OSI to OMG experiences from the port of na ISODE based application to OMG Corba concepts. **Computer Communications**, v. 19, p. 4-12, 1996.
- [Eriksson, 93] Eriksson, I.; McFadden, F. Quality function deployment: a tool to improve software quality. **Information and Software Technology**, v. 35, n. 9, p. 491-98, 1993.
- [Farooqui, 95] Farooqui, K.; Logrippo, L.; Meer, J. The ISO reference model for open distributed processing: an introduction. **Computer Networks and ISDN Systems**, v. 27, p. 1215-29, 1995.
- [Farooqui, 96] Farooqui, K.; Logrippo, L. Group communication models. **Computer Communications**, v. 19, p. 1276-88, 1996.
- [Fatoohi, 97] Fatoohi, R.A. Performance evaluation of communication software systems for distributed computing. **The British Computer Society**, v. 4, p. 169-75, 1997.
- [Gale, 92] Gale, S. Desktop video conferencing: technical advances and evaluation issues. **Computer Communications**, v. 15, n. 8, p. 517-25, 1992.
- [Garlan, 95] Garlan, D.; Perry, D.E. Introduction to the special issue on software architecture. **IEEE Transaction on Software Engineering**, v. 21, n. 4, p. 269-74, 1995.
- [Garrahan, 93] Garrahan, J.J.; Russo, P.A.; Kitami, K.; Kung, R. Intelligent network overview. **IEEE Communications Magazine**, p. 30-6, 1993.
- [Gatenby, 94] Gatenby, D.A.; Lee, P.M.; Howard, R.E.; Hushyar, K.; Layendecker, R.; Wesner, J. Concurrent engineering: an enabler for fast, high-quality product realization. **AT&T Technical Journal**, p. 34-46, 1994.

- [Gay, 95] Gay, V.; Leydekkers, P.; Veld, R.H. Specification of multiparty audio and interaction based on the reference model of open distributed processing. **Computer Networks and ISDN**. [online]. p. 1-17. Available: http://www.laria.u-picardie.fr/~gay/html/body_publications.html, february 1998.
- [Geihs, 93] Geihs, K, Mann, A. ODP viewpoints of IBCN service management. **Computer Communications**, v. 16, n. 11, p. 695-705, 1993.
- [Gelman, 96] Gelman, S.J.; Peck, W.D. Bringing business information to AT&T network systems through a data warehouse. **AT&T Technical Journal**, p. 68-78, 1996.
- [Gotzhein, 95] Gotzhein, R. Towards a basic reference model ODP. **Computer Networks and ISDN Systems**, v. 27, p. 1287-304, 1995.
- [Guerraoui, 97] Guerraoui, R.; Vinoski, S. Special issue on Corba. **The British Computer Society**, v. 4, p. 129, 1997.
- [Handfield, 94] Handfield, R. Effects of concurrent engineering on make-to-order products. **IEEE Transaction on engineering management**, v. 41, n. 4, p. 384-93, 1994.
- [Hehmann, 90] Hehmann, D.B.; Salmony, M.G.; Stüttgen, H.J. Transport services for multimedia applications on broadband networks. **Computer Communications**, v. 13, n. 4, p. 197-203, 1990.
- [Heinrichs, 93] Heinrichs, B.; Jakobs, K.; Carone, A. High performance transfer services to support multimedia group communications. **Computer Communications**, v. 16, n. 9, p. 539-47, 1993.
- [Herbert, 88] Herbert, A.; Dobson, J.; Monk, J.; Linden, R. A design schema for distributed systems. **Computer Science**, p. 697-707, 1988.
- [Herbert, 94] Herbert, A. An ANSA overview. **IEEE Networks**, p. 18-23, 1994.
- [Hirama, 95] Hirama, K. **Método de desenvolvimento de sistemas orientado a objetos: Uma abordagem integrada à análise estruturada e redes de petri**. São Paulo, 1995.

- Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo.
- [Hollingsworth, 94] Hollingsworth, D. Document TC00-1003 issue 1.1: workflow management coalition: the workflow reference model, 1994.
- [Hunter, 89] Hunter, R.; Kaijser, P.; Nielsen, F. ODA: a document architecture for open systems. **Butterworth & Co**, v. 12, n. 2, p. 69-79, 1989.
- [IEEE, 93] IEEE, **IEEE guide to software requirements specifications**, The institute of electrical and electronics engineers, 1993.
- [ISO, 94] ISO/IEC 9646-1: information technology – open system interconnection conformance testing methodology and framework – part 1 : general concept, 1994.
- [ISO, 95 a] ISO. Recommendation X.901/ISO/IEC 10746-1: basic reference model of open distributed processing - part 1: overview and guide to use, 1995.
- [ISO, 95 b] ISO. Recommendation X.902/ISO/IEC 10746-2: information technology - open distributed processing - reference model: foundations, 1995.
- [ISO, 95 c] ISO. Recommendation X.903/ISO/IEC 10746-3: information technology - open distributed processing - reference model: architecture, 1995.
- [ISO, 95 d] ISO. Recommendation X.904/ISO/IEC 10746-4: | basic reference model of open distributed processing - part 4: architectural semantics amendment, 1995.
- [ISO, 95 e] ISO. Information technology - open distributed processing - ODP trading function - editors draft DIS text, 1995.
- [ITU, 92] ITU. Recommendation M.3010: principle for telecommunication management network, 1992.
- [ITU, 96] ITU. Recommendation G.851-01: management of the transport network-application of the RM-ODP framework, 1996.
- [Jacobson, 92] Jacobson, I.; Christerson, M.; Jonsson, P.; Overgaard, G. **Object-oriented software engineering - a use case driven approach**. Reading MA: Addison Wesley, 1992.

- [Jordan, 91] Jordan, K.A.; Davis, A.M. Requirements engineering metamodel: an integrated view of requirements. In: COMPASAC 91 - The 15 th Annual Conference Computer Software. **Anais**. p. 472-78, 1991.
- [Joyner, 97] Joyner, I. Open distributed processing: unplugged. [online]. p. 1-22. Available:<http://homepages.tig.com.au/~ijoyner/ODPUnplugged.html>, february 1998.
- [Junqueira, 95] Junqueira, G.B. Da Engenharia tradicional à engenharia simultânea no setor industrial nacional. **Boletim da Escola Politécnica da Universidade de São Paulo**. p. 1-18, 1995. /Separata/
- [Kamel, 92] Kamel, M.N.; Kamel, N.N. Federated database management system: requirements, issues and solutions. **Computer Communications**, v. 15, n. 4, p. 270-78, 1992.
- [Kaplan, 92] Kaplan, S.M.; Carroll, A.M. Supporting collaborative processes with conversation builder. **Computer Communications**, v. 15, n. 8, p. 489-501, 1992.
- [Karandikar, 92] Karandikar, H.M.; Wood, R.T.; Byrd Jr., J. **Process and technology readiness assessment for implementing concurrent engineering**. Morgantown, CERC-TR-TM-92-004, 1992. (CERC Technical Report Series. Technical Memoranda, 1-9)
- [Karinthi, 92 a] Karinthi, R. et al. **Promoting concurrent engineering through information sharing**. Morgantown, CERC-TR-TM-92-010, 1992. (CERC Technical Report Series. Technical Memoranda, 1-15)
- [Karinthi, 92 b] Karinthi, R.; Jagannathan, V.; Montan, V.; Petro, J.; Raman, R.; Trapp, G. Integrating heterogeneous information repositories in a concurrent engineering environment. In: **WORKSHOP ON ENABLING TECHNOLOGIES FOR CONCURRENT ENGINEERING**, 1., Morgantown, 1992. **Proceedings**. WV, 1992. v. 1, p. 1-15.
- [Kervella, 94 a] Kervella, B.; Gay, V.; Horlait, E. Integration of a scenario service in a multimedia messaging system. **ISMM International Conference on distributed multimedia systems and applications**. [online]. p. 1-12. Available: http://www.laria.u-picardie.fr/~gay/html/body_publications.html, february 1998.

- [Kervella, 94 b] Kervella, B.; Gay, V.; Horlait, E. Towards a complete multimedia mail: use of MHE in standard messaging systems. **Conference on Multimedia Transport and Teleservices**. [online]. p. 1-11. Available: http://www.laria.u-picardie.fr/~gay/html/body_publications.html, february 1998.
- [Kilov, 93] Kilov, H. Precise specification of behavior in object-oriented standardization activities. **Computer Standards and Interfaces**, v. 15, n. 2-3, p. 275-85.
- [Kilov, 94] Kilov, H.; Redmann, L. **Information modeling: an object-oriented approach**. 1.ed. New Jersey, Prentice-Hall, 1994.
- [Kilov, 95] Kilov, H.; Cuthbert, L. A model for document management. **Computer Communications**, v. 16, n. 6, p. 408-17, 1995.
- [Kiname, 96] Kiname, B.; Muldowney, D. Distributing broadband multimedia systems using Corba. **Computer Communications**, v. 19, p. 13-21, 1996.
- [Klein, 93] Klein, M. Capturing design rationale in concurrent engineering teams. **Computer**, v. 26, n. 1, p. 39-47, 1993.
- [Kruglianskas, 93] Kruglianskas, I. Engenharia simultânea: organização e implantação em empresas brasileiras. **Revista de Administração, São Paulo**, v. 28, n. 4, p. 104-10, 1993.
- [Kythe, 96] Kythe, D.K. The promise of distributed business components. **AT&T Technical Journal**, p. 20-8, 1996.
- [LaBauve, 92] LaBauve, L. Lessons learned in the implementation of concurrent engineering. **Concurrent Engineering Research in Review**, v. 4, p. 23-8, 1992. Special issue.
- [Lann, 92] Lann, G.L. Designing real-time dependable distributed systems. **Computer Communications**, v. 15, n. 4, p. 225-34, 1992.
- [Larman, 97] Larman, C. **Applying UML and patterns: an introduction to object-oriented analysis and design**. 1.ed. New Jersey, Prentice Hall PTR, 1997.
- [Lawson, 93] Lawson, M. **Justifying CE**. Morgantown, CERC-TR-RN-93-70, 1993. (CERC Technical Report Series. Research Note, 30)

- [Leydekkers, 94] Leydekkers, P.; Gay, V. Multimedia conferencing services in an open distributed environment. **2nd IEEE International workshop on advanced teleservices and high speed communication architectures**. [online]. p. 1-14. Available: http://www.laria.u-picardie.fr/~gay/html/body_publications.html, february 1998.
- [Leydekkers, 96] Leydekkers, P.; Gay, V. ODP view on quality of service for open distributed multimedia environments. **IWQoS'96**. [online]. p. 1-13. Available: http://www.laria.u-picardie.fr/~gay/html/body_publications.html, february 1998.
- [Li, 95] Li, G. An overview of real-time ANSAware 1.0. **Distributed System Engineering**, v. 2, p. 28-38, 1995.
- [Liang ,97] Liang, J.; Sédillot, S.; Traverson, B. Object management group object transaction service based on an X/Open and internacional organization for standardization open systems interconnection transaction processing kernel. **The British Computer Society**, v. 4, p. 151-9, 1997.
- [Liang, 94] Liang, T.; Lai, H.; Chen, N. When client/server isn't enough: coordinating multiple distributed tasks. **Computer**, v. 27, n. 5, p. 73-9, 1994.
- [Lins, 97] Lins, A.; Ribeiro, A.D.; Carvalho, D.M.L. **Integração de ambiente de desenvolvimento utilizando recursos de intranet para aplicação de engenharia simultânea**. São Paulo, 1997. Trabalho de formatura (Graduação) – Escola Politécnica, Universidade de São Paulo.
- [Londono, 92] Londono, F.; Cleetus, K.J.; Nichols, D.M.; Iyer, S.; Karandikar, H.M.; Reddy, S.M.; Potnis, S.M.; Massey, B.; Reddy, A.; Ganti, V. **Coordinating a virtual team**. Morgantown, CERC-TR-RN-92-005, 1992. (CERC Technical Report Series. Research Note, 30)
- [Macartney, 92] Macartney, A.J.; Blair, G.S. Flexible trading in distributed multimedia systems. **Computer Networks and ISDN Systems**, p. 145-57, 1992.
- [Magedanz, 93] Magedanz, T. IN and TMN: the basis for future information networking architectures. **Computer Communications**, v. 16, n. 5, p. 267-76, 1993.

- [Maliniak, 91] Maliniak, L. Teamwork is the key to concurrent desing. **Electronic Desing**, p. 37-50, 1991.
- [Martucci, 92] Martucci Jr., M. **Estudos de estruturas de sistemas de automação**. São Paulo, 1992. Tese (Livre Docência) - Escola Politécnica, Universidade de São Paulo.
- [Matsuyama, 97] Matsuyama, F. **Generalização dos níveis de controle e instrumentação de um sistema aberto de automação**. São Paulo, 1997. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo.
- [Meer, 95] Meer, J. The ISO reference model for open distributed processing. **Computer Networks and ISDN Systems**, v. 27, p. 1211-14, 1995.
- [Muftic, 94] Muftic, S. Security architecture for ODP systems Final results of the CEC COST-11 Ter "security" project. **Computer Networks and ISDN Systems**, v. 26, p. 1343-49, 1994.
- [Mulvihill, 93] Mulvihill, C.; McDermott, G.; Patel, A. Cooperative decision support for medical diagnosis. **Computer Communications**, v. 16, n. 9, p. 581-93, 1993.
- [Muños, 98] Muños, G.T. **Comparação de arquitetura centralizada com arquitetura distribuída no nível de instrumentação**. São Paulo, 1990. 142p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo.
- [Najm, 95] Najm, E.; Stefani, J. A formal semantics for the ODP computacional model. **Computer Networks and ISDN Systems**. v. 27, p. 1305-29, 1995.
- [Nankman, 96] Nankman, M.; Lambert, J.M. Specification of a distributed storage system. **Computer Communications**, v. 19, p. 30-8, 1996.
- [Narasimhan, 97] Narasimhan, P.; Moser, L.E.; Melliar-Smith, P.M. Replica consistency of Corba objects in partitionable distributed systems. **The British Computer Society**, v. 4, p. 139-50, 1997.
- [Navarro, 93] Navarro, L.; Prinz, W.; Rodden, T. CSCW requires open systems. **Computer Communications**, v. 16, n. 5, p. 288-97, 1993.

- [Nehmer, 92] Nehmer, J.; Mattern, F. Framework for the organization of cooperative services in distributed client-server systems. **Computer Communications**, v. 15, n. 4, p. 261-77, 1992.
- [Nicol, 93] Nicol, J.R.; Wilkes, C. T.; Manola, F.A. Object orientation in heterogeneous distributed computing systems. **Computer**, p. 57-67, 1993.
- [OMG 95] Object management group 1995 The Common Object Request Broker: Architecture and specification revision 2.0.
- [Palme, 92] Palme, J.; Tholerus, T. SuperKOM - design considerations for a distributed, highly structured computer conferencing system. **Computer Communications**, v. 15, n. 8, p. 509-16, 1992.
- [Palme, 93] Palme, J. Standards for asynchronous group communication. **Computer Communications**, v. 16, n. 9, p. 532-8, 1993.
- [Palmer, 91] Palmer, B.; Korbly, L.; Trapp, G. **Modeling the concurrent engineering process**. Morgantown, CERC-TR-RN-91-010, 1991. (CERC Technical Report Series. Research Note, 1-15)
- [Palmer, 94] Palmer, J.D.; Fields, N.A.; Brouse, P.L. Multigroup decision-support systems in CSCW. **Computer**, v. 27, n. 5, p. 67-72, 1994.
- [Proctor, 97] Proctor, F.M.; Albus, J.S. Open-architecture controllers. **IEEE Spectrum**, p. 60-3, 1997.
- [Rahkila, 97] Rahkila, S.; Stenberg, S. Experiences on integration of network management and a distributed computing platform. **The British Computer Society**, v. 4, p. 130-8, 1997.
- [Rakos, 90] Rakos, J.J. **Software project management for small to medium sized projects**. 1.ed. New Jersey, Prentice-Hall, 1990.
- [Ramana, 93] Ramana, Y.V.; Karinthy, R.; Kanakanahalli, S. Computer support for concurrent engineering. **Computer**, v. 26, n. 1, p. 12-15, 1993.
- [Rational, 98] RATIONAL SOFTWARE CORPORATION. Management with requisitepro. [online]. p. 1-6. Available: <http://www.rational.com/>, february 1998.

- [Reddy, 92] Reddy, Y.V.R.; Wood, R.T.; Cleetus, K.J. The DARPA initiative in concurrent engineering. **Concurrent Engineering Research center**, v. 1, p. 2-10, 1992.
- [Reddy, 93] Reddy, Y.V.R.; Srinivas, K.; Jagannathan, V.; Karinthy, R. Computer support for concurrent engineering. **Computer**, v. 26, n. 1, p. 12-16, 1993.
- [Reinhard, 94] Reinhard, W.; Schweitzer, J.; Völksen, G. CSCW tools: concepts and architectures. **Computer**, v. 27, n. 5, p. 28-36, 1994.
- [Robinson, 91] Robinson, D. Remote procedure call: a stepping stone towards ODP. **Computer Networks and ISDN Systems**, v. 23, p. 191-4, 1991.
- [Rodden, 92] Rodden, T.; Blair, G.S. Distributed systems support for computer supported cooperative work. **Computer Communications**, v. 15, n. 8, p. 527-38.
- [Rosa, 95] Rosa, P.F. **Um ambiente para desenvolvimento de sistemas distribuídos**. São Paulo, 1995. 167p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo.
- [Rosenberry, 92] Rosenberry, W.; Kenney, D.; Fisher, G. **Understanding DCE**. 1.ed. O'Reilly & Associates, 1992.
- [Rubin, 94] Rubin, H.; Natarajan, N. A distributed software architecture for telecommunication network. **IEEE Network**, p. 8-17, 1994.
- [Rumbaugh, 91] Rubin, H.; Natarajan, N. **Object Oriented Modeling and Design**. 1991.
- [Ryan, 92] Ryan, J.C.; Jons, O.P. Improving the ship design, acquisition and construction process. **Naval Engineers Journal**, p. 39-57, 1992.
- [Scherer, 97] Scherer, D.; Murer, T.; Würtz, A. Designing the distributed architecture DIPS for cooperative software engineering. **The British Computer Society**, v. 4, p. 160-8, 1997.
- [Schill, 90] Schill, A. Distributed application support: survey and synthesis of existing approaches. **Information and Software Technology**, v. 32, n. 8, p. 545-58, 1990.

- [Schill, 91] Schill, A. Distributed system and execution model for office environments. **Computer Communications**, p. 478-88, 1991.
- [Shaw, 96] Shaw, M.; Garlan, D. **Software architecture: perspectives on an emerging discipline**. 1.ed. New Jersey, Prentice-Hall, 1996.
- [Shiratori, 96] Shiratori, N.; Suganuma, T.; Sugiura, S.; Chakraborty, G.; Sugawara, K.; Kinoshita, T.; Lee, E.S. Framework of a flexible computer communication network. **Computer Communications**, p. 1268-75, 1996.
- [Siegel, 96] Siegel, J. **Corba fundamentals and programming**. 1.ed. John Wiley & Sons, 1996.
- [Silva, 96] Silva, M.M. **Proposta de um sistema para integração e desenvolvimento de projetos através de engenharia simultânea**, São Paulo, 1996. 164p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo.
- [Sinderen, 95] Sinderen, M.; Ferreira, L.P.; Vissers, C.A.; Katoen, J. A design model for open distributed processing system. **Computer Networks and ISDN Systems**, v. 27, p. 1263-85, 1995.
- [Soares, 95] Soares, L.F.G.; Lemos, G.; Colcher, S. **Redes de computadores: das LANs, MANs e WANs às redes ATM**. 2.ed. Rio de Janeiro, Campus, 1995.
- [Spina, 98] Spina, E. **Um método para a melhoria da qualidade de fornecimentos da pequena empresa de eletrônica**. São Paulo, 1998. 264p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo.
- [Sprague, 91] Sprague, R.A.; Singh, K.J. Concurrent engineering in product development. **IEEE Desing & Test of Computers**, p. 6-13, 1991.
- [Sriram, 92] Sriram, R.; Livezey, B.; Perkins, W. A distributed shared database system for concurrent engineering. In: **WORKSHOP ON ENABLING TECHNOLOGIES FOR CONCURRENT ENGINEERING**, 1., Morgantown, 1992. **Proceedings**. Concurrent Engineering Research Center, 1992. p. 29-43.

- [Stefani, 95] Stefani, J. Open distributed processing: an architectural basis for information networks. **Computer Communications**, v. 18, n. 11, p. 849-62, 1995.
- [Stevens, 87] Steven, R.J.; Greenwood, R.T.; Alvisi, G.; Deschamps, M. Design of Large Distributed Information System. **Computer Networks and ISDN System**, n. 14, p. 123-35, 1987.
- [Tanenbaum, 95] Tanenbaum, A.S. **Distributed operating systems**. 1.ed. New Jersey, Prentice-Hall, 1995.
- [Tanenbaum, 97] Tanenbaum, J.M.; Chowdhry, T.S.; Hughes, K. Eco system: an internet commerce architecture. **Computer**, v. 30, n. 5, p. 48-55, 1997.
- [Tanomaru, 94] 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.
- [Tschammer, 94] Tschammer, V.; Magedanz, T.; Tschichholz, M.; Wolisz, A. Cooperative management in open distributed systems. **Computer Communications**, v. 17, n. 10, p. 717-28, 1994.
- [Tschichholtz, 96] Tschichholtz, M.; Tschammer, V.; Dittrich, A. Integrated approach to open distributed management. **Computer Communications**, v. 19, p. 76-87, 1996.
- [Turban, 98] Turban, E.; Aronson, J.E. **Decision support systems and intelligent systems**. 5.ed. New Jersey, Prentice-Hall, 1998.
- [UML, 97] UML Sumary. p. 1-269. Available: <http://www.rational.com>, 1997.
- [Valérie, 95] Valérie, G.; Leydekkers, P.; Huis, R. Specification of multiparty audio and video inteaction based on the reference model open distributed processing. **Computer Network and ISDN System**, v. 27, p. 1247-62, 1995.
- [Wolisz, 93] Wolisz, A.; Tschammer, V. Performance aspects of trading in open distributed systems. **Computer Communications**, v. 16, n. 5, p. 277-87, 1993.