

Este exemplar refere-se à redação final da
Tese/Dissertação devidamente corrigida e defendida
por: Carlos Alberto Previdelli
e aprovada pelo Banco Examinadora.
Campinas, 19 de Junho de 2002
COORDENADOR DE PÓS-GRADUAÇÃO

**Migração de Sistemas Legados para Ambiente
de Suporte a Projetos e Operação**

Carlos Alberto Previdelli

Dissertação de Mestrado

Migração de Sistemas Legados para Ambiente de Suporte a Projetos e Operação

Carlos Alberto Previdelli¹

Dezembro de 2001

Banca Examinadora:

- Prof. Dr. Geovane Cayres Magalhães
Instituto de Computação, Unicamp (Orientador)
- Prof. Dr. José Alberto Quintanilha
Escola Politécnica/USP
- Profa. Dra. Cláudia M. Bauzer Medeiros
Instituto de Computação, Unicamp
- Prof. Dr. Luis Marcos Gonçalves
Instituto de Computação, Unicamp (Suplente)

¹O desenvolvimento deste trabalho foi parcialmente custeado com recursos do Funttel - Fundo para o Desenvolvimento Tecnológico das Telecomunicações, repassado à Fundação CPqD através da autorização da Portaria no. 581 de 08/10/2001

UNIDADE 30
Nº CHAMADA II UNICAMP
P929m
V _____ EX _____
TOMBO BCI 50162
PROC 16.837/02
C _____ DX _____
PREÇO R\$ 11,00
DATA 31/07/02
Nº CPD _____

CMO0171056-5

BIB ID 249046

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Previdelli, Carlos Alberto

P929m Migração de sistemas legados para ambiente de suporte a projetos e operação / Carlos Alberto Previdelli -- Campinas, [S.P. :s.n.], 2001.

Orientador : Geovane Cayres Magalhães

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Engenharia de software. 2. Banco de dados. 3. Sistemas de informação geográfica. I. Magalhães, Geovane Cayres. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Migração de Sistemas Legados para Ambiente de Suporte a Projetos e Operação

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Carlos Alberto Previdelli e aprovada pela Banca Examinadora.

Campinas, 21 de dezembro de 2001.



Prof. Dr. Geovane Cayres Magalhães
Instituto de Computação, Unicamp
(Orientador)

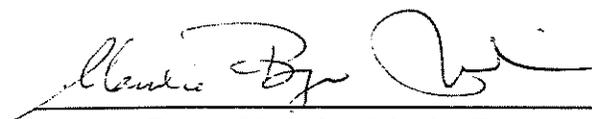
Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

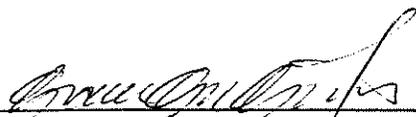
Tese defendida e aprovada em 21 de dezembro de 2001, pela
Banca Examinadora composta pelos Professores Doutores:



Prof. Dr. José Alberto Quintanilha
USP



Prof. Dra. Claudia Maria Bauzer Medeiros
IC - UNICAMP



Prof. Dr. Geovane Cayres Magalhães
IC - UNICAMP

© Carlos Alberto Previdelli, 2002.
Todos os direitos reservados.

Resumo

Sistemas de Informação são imprescindíveis nas grandes corporações. Particularmente nas áreas de suporte a projeto e operações de empresas de telecomunicações, os sistemas de informação são a grande vantagem competitiva. Estes sistemas devem ser flexíveis, estáveis e confiáveis para suportar grandes volumes de transações em um cenário de missão crítica. Este, porém, não é o ambiente mais comum nas grandes empresas. A quantidade de sistemas inflexíveis, sem documentação, com dados pouco confiáveis e com alto custo de evolução e manutenção é grande. A saída para este impasse está na migração destes sistemas legados para um ambiente de sistemas de informação com arquitetura aberta, ambiente distribuído e que possam ser utilizados por todas as áreas, estando integrados com os demais sistemas da empresa. A migração destes grandes sistemas legados de uma forma abrupta seria um fracasso principalmente em virtude do desconhecimento dos sistemas legados. A migração incremental destes sistemas é a mais apropriada nestes casos. Esta dissertação apresenta uma metodologia para migração de sistemas legados para ambientes de suporte a projetos e operação em etapas. A metodologia desenvolvida foi validada num caso prático no contexto do SAGRE - um sistema de suporte a operações, projetos e informações geográficas desenvolvido pelo CPqD.

Abstract

Information Systems are essential for corporations. Particularly at design and operational support areas of telecommunications companies, information systems are the competitive advantage. These systems must be flexible, stable and reliable in order to support a great volume of transactions in a mission critical scenario. This is not the common environment within corporations. There is an expressive amount of inflexible, nondocumented, unreliable systems which bear costly maintenance and evolution. The solution for this situation is the migration of these legacy systems into a new integrated and distributed environment that can also be used by other corporate systems. The abrupt migration of these systems could not be successful due to a lack of knowledge about them. The incremental approach is more appropriate in these cases. This dissertation shows a methodology for migrating legacy systems that support design and operational environments in an incremental mode. The methodology is validated by a case study within the context of SAGRE - a system which support geographic operations, projects and information, developed by CPqD.

À minha esposa, aos meus filhos e aos meus pais

Agradecimentos

O amor jamais acaba; mas, havendo profecias, desaparecerão; havendo línguas, cessarão; havendo ciência passará - 1 Coríntios 13:8

Agraceço ao Senhor pelas oportunidades que me deu durante toda esta etapa e pelos amigos dentro e fora do Instituto de Computação.

Agraceço especialmente à minha esposa Tatiana que sempre me apoiou com seu amor e compreensão.

Ao meu filho Pedro Henrique pelos momentos que lhe foram negados.

Aos meus pais Antenor e Inês meus agradecimentos pelo apoio e encorajamento que foram essenciais para a conclusão deste trabalho.

Ao orientador e amigo Geovane Cayres Magalhães agradeço pelas oportunidades oferecidas e ao estímulo constante.

Aos meus sogros Francisco e Daniela pelos cuidados dispensados à minha família.

Ao amigo Mário Harada que sempre me estimulou a concluir este trabalho e pelo detalhado trabalho de revisão.

Aos amigos que me auxiliaram na revisão gramatical: Andreneison e Yuri.

Agradeço aos professores e a todos os funcionários do Instituto de Computação.

Sumário

1	Introdução	1
2	Conceitos Básicos	5
2.1	Bancos de Dados Heterogêneos	5
2.1.1	Heterogeneidade, Autonomia e Distribuição (HAD)	8
2.1.2	Abordagens para Integração de SBDH	9
2.2	Sistemas de Informações Legados	16
2.2.1	Estratégia <i>Cold Turkey</i>	18
2.2.2	Estratégia <i>Chicken Little</i>	18
2.2.3	Trabalhos Correlatos	19
2.3	Mapeamento dos Processos de Negócio	23
2.4	Arquitetura OMG	28
2.4.1	O Modelo de Objetos do OMG	29
2.4.2	OMA	31
2.5	Resumo do Capítulo	33
3	Metodologia de Migração de Sistemas Legados	35
3.1	Metodologia Proposta	35
3.2	Análise Global	39
3.2.1	Mapeamento dos Processos	39
3.2.2	Mapeamento dos Sistemas de Informações	43
3.2.3	Análise dos Processos e Sistemas Legados	44
3.2.4	Priorização dos Processos e Sistemas Legados	44
3.3	Arquitetura de Sistemas de Informação	45
3.3.1	Arquitetura de Sistemas Legados	46
3.3.2	Características de Arquitetura de Sistemas de Informação	48
3.4	Migração Incremental	59
3.5	Resumo do Capítulo	63

4	Estudo de Caso da Metodologia	65
4.1	Projeto SAGRE	65
4.2	Sistemas Legados na EOT	67
4.3	Análise Global	68
4.3.1	Mapeamento dos Processos	68
4.3.2	Mapeamento dos Sistemas de Informações	79
4.3.3	Análise dos Processos	79
4.3.4	Priorização dos Processos e Sistemas Legados	81
4.4	Arquitetura de Sistemas de Informações para a EOT	82
4.4.1	Arquitetura dos Sistemas Legados	83
4.4.2	Arquitetura dos novos Sistemas Informação	84
4.5	Migração dos Sistemas Legados	95
4.5.1	Detalhar o Sistema Legado	95
4.5.2	Projetar e Construir o Sistema de Informação Destino	98
4.5.3	Projetar e Construir Ferramentas para Conversão dos Dados do Sistema Legado para o Sistema Destino	107
4.5.4	Definir o Plano de Implantação do Ambiente Final e Migração do Sistema Legado	109
4.6	Resumo do Capítulo	110
5	Conclusões	113
	Bibliografia	117

Lista de Tabelas

2.1	Grau de Acoplamento entre SBDH	7
2.2	Classificação de Processos	27
2.3	Avaliação de Arquiteturas quanto a Distribuição	28
3.1	Matriz de Mapeamento: Processos X Sistemas	44
4.1	Matriz de Mapeamento Processos X Sistemas da EOT	79
4.2	Matriz de classificação dos dados de Caixa Terminal	98

Lista de Figuras

2.1	Arquitetura de Referência para SBDH [SL90]	11
2.2	Transparência de Modelos	12
2.3	Processamento de Consultas Globais	13
2.4	Arquitetura de Mediadores em três Camadas	15
2.5	Grafo de Dependências entre os Passos da Migração	20
2.6	<i>Object Management Architecture</i>	32
3.1	Metodologia para Migração de Sistemas Legados	38
3.2	Arquitetura de Sistemas de Informação	47
3.3	<i>Forward and Reverse gateways</i>	49
4.1	Casos de Uso de Atendimento a Clientes	69
4.2	Casos de Uso de Engenharia	71
4.3	Modelo de Objetos de Atendimento a Clientes	74
4.4	Modelo de Objetos de Construção de Rede	75
4.5	Modelo de Objetos de Projeto	76
4.6	Modelo de Objetos de Mercado	77
4.7	Arquitetura dos sistemas legados SGC e SGF	84
4.8	Arquitetura de Rede	85
4.9	Arquitetura de Integração SAGRE e SGC	86
4.10	Composição hierárquica dos componentes do SAGRE	89
4.11	Arquitetura GAT	91
4.12	Arquitetura cache dados gráficos	92
4.13	Arquitetura de componentes do SAGRE	94
4.14	Modelo de Dados do SGF	97
4.15	Diagrama de Sequência do fluxo de Abertura e Fechamento de OS	101
4.16	Tela do SAGRE/Oper	103
4.17	Arquitetura dos subsistemas do módulo de operação	104
4.18	Modelo de dados - SAGRE/Oper	105
4.19	Modelo de Dados SAGRE/Oper	106

Capítulo 1

Introdução

A competitividade entre as grandes organizações, a constante necessidade de aumento da produtividade e a crescente automação nos processos de negócio, aliado à redução no custo de equipamentos e rede de comunicação têm levado as empresas a dependerem cada vez mais das informações armazenadas em seus bancos de dados. As informações precisam estar atualizadas, precisas e disponíveis em qualquer lugar da organização ou até fora dela. Surge então a necessidade de integrar os sistemas e os dados de toda a empresa. Os sistemas de informação não são mais utilizados apenas pela equipe de analistas de sistemas. Eles devem estar espalhados por toda a organização e serão utilizados por qualquer pessoa, desde o diretor até o pessoal da linha de produção. Desta forma toda a complexidade do sistema e a estrutura interna dos dados devem ser transparente para o usuário.

Atualmente os grandes sistemas de informações legados são um problema dominante entre várias organizações ou grandes indústrias. Um dos principais problemas enfrentados pelos sistemas legados é a arquitetura com que eles foram construídos, gerando dificuldades na manutenção dos sistemas para que eles possam continuar atendendo às necessidades dos negócios. Não é possível, na maioria dos casos, separar a interação com usuários, o acesso aos dados, as regras de negócio e a interface com outros sistemas. Todo o código geralmente está construído de uma forma não estruturada ou monolítica. Outro problema enfrentado é a dificuldade em deixar os sistemas legados operando, devido principalmente à defasagem tecnológica e à falta ou dificuldade de integração destes sistemas com os novos sistemas que estão sendo desenvolvidos ou adquiridos pela organização. A melhor saída para estes casos é a migração dos sistemas legados para novos sistemas baseados em uma arquitetura aberta e flexível.

O rápido desenvolvimento da tecnologia da informação tem gerado uma heterogeneidade cada vez maior dos sistemas que estão sendo desenvolvidos, além dos sistemas legados existentes. Para que os novos sistemas que estão sendo construídos não se tornem os

sistemas legados do futuro, eles devem ser desenvolvidos considerando a possibilidade de atualização, mudança de *hardware*, mudança dos *softwares* que o compõem, e a possibilidade de integração com outros sistemas. Para desenvolver este tipo de sistema, deve ser projetada uma arquitetura baseada em componentes que sejam interoperáveis, portáteis e flexíveis. Segundo Brodie et al. [BS95], Wegner [Weg96] e Silberschatz et al. [SZ96] em grandes organizações não existirá apenas uma arquitetura mas uma combinação de arquiteturas e o requisito básico é a interoperação transparente entre elas.

Exemplo desta heterogeneidade são os sistemas das Empresas Operadoras de Telecomunicações (EOT). Elas têm utilizado sistemas de informação por muitos anos. Estes sistemas controlam desde as chamadas telefônicas feitas na central telefônica até informações e atendimento sobre clientes, faturamento e gerência de rede. Vários destes sistemas são de missão crítica¹ tornando a sua migração muito mais complexa.

Durante o processo de migração surgem inúmeros desafios técnicos, como por exemplo: integração de sistemas, integração de dados heterogêneos, metodologias para a migração de sistemas e definição de uma arquitetura para sistemas distribuídos.

O objetivo desta dissertação é integrar algumas das principais metodologias de migração de sistemas legados existentes na literatura para definir uma metodologia genérica de migração de sistemas legados e posteriormente validar a metodologia integrada sobre um caso real dentro de um ambiente de projeto e operação como das EOT. Para fazer a migração deste sistema e também como parte da metodologia de migração serão propostas características de uma arquitetura de sistemas de informações. Estas características auxiliarão na definição dos novos sistemas para os quais os sistemas legados serão migrados. Para propor as características desta arquitetura, duas tecnologias serão analisadas: sistemas de banco de dados heterogêneos (SBDH) e integração de sistemas. Apesar destes tópicos não serem o objetivo principal da dissertação, será feita uma revisão dos conceitos básicos destes assuntos, uma vez que auxiliarão na definição da arquitetura de migração.

O SAGRE (Sistema Automatizado de Gerência de Rede Externa) será utilizado como estudo de caso desta dissertação. Ele está em operação e integrado com vários sistemas legados existentes nas EOT brasileiras e é um sistemas de missão crítica para estas empresas. Apesar do SAGRE utilizar um sistema de informações geográficas (SIG) como parte da sua arquitetura, os problemas específicos de integração de sistemas legados a SIG não serão abordados por ser um tópico muito amplo e estar fora do escopo desta dissertação.

As principais contribuições deste trabalho são:

- definição de uma metodologia para migração de sistemas legados para sistemas baseados em plataformas abertas e distribuídas;

¹sistemas vitais para o funcionamento das empresas e a sua paralisação representa a perda substancial de receita

- Aplicação da metodologia em um ambiente de suporte a projeto e operação com grandes sistemas legados.

A dissertação está organizada da seguinte forma. O capítulo 2 traz uma revisão dos conceitos básicos sobre mapeamento de processos de negócio, sistemas legados, sobre a arquitetura CORBA (*Common Object Request Broker Architecture*) e sobre SBDH. O capítulo 3 apresenta a proposta da metodologia para migração dos sistemas legados. No capítulo 4 é feita a validação da metodologia através de um estudo de caso utilizando o SAGRE e por fim, o capítulo 5 resume as conclusões e extensões desta dissertação.

No capítulo 4, onde é realizada uma aplicação prática da metodologia, nem todos os materiais e produtos gerados são descritos por extrapolarem o escopo deste trabalho. Também alguns nomes e informações são resumidos por questões de sigilo das informações. Porém o estudo de caso não foi afetado por estes fatores limitantes.

Capítulo 2

Conceitos Básicos

Neste capítulo é feita uma revisão dos principais conceitos relativos à heterogeneidade dos bancos de dados, são caracterizados os sistemas de informações legados, levantado os aspectos necessários para o mapeamento de processos de negócio e apresentada a arquitetura da OMG (*Object Management Group*) utilizada na construção de sistemas distribuídos [CDK94]. Todos estes tópicos são relevantes para o processo de migração de sistemas legados e construção de sistemas de informação que são os objetivos desta dissertação. A partir deste capítulo os termos mediadores, *gateway* e *middleware* serão utilizados indistintamente.

2.1 Bancos de Dados Heterogêneos

Historicamente os sistemas de banco de dados (SBD) apareceram por uma necessidade de armazenamento, acesso e manipulação dos dados de uma maneira compartilhada e segura em relação aos tradicionais sistemas de arquivos. Com os SBD, tornou-se possível uma maior independência das aplicações sobre a organização física dos dados, diminuição das redundâncias e conseqüentemente diminuição das inconsistências, tornando os dados mais seguros e íntegros.

Os SBD tradicionais são compostos pelas bases de dados e por um sistema gerenciador de banco de dados (SGBD) que é o *software* que controla e permite operações sobre os dados armazenados.

Os modelos lógicos são usados para representar as informações ou a semântica de um determinado domínio do mundo real para, posteriormente, serem mapeadas para o modelo do SGBD gerando um esquema que será utilizado como ferramenta para a construção das aplicações.

Os primeiros modelos utilizados foram orientados a registros, onde os dados são representados como uma sequência fixa de campos valorados chamadas registros. Cada tipo

de registro tem os mesmos tipos de campos. Exemplos destes modelos são os modelos hierárquicos, de rede e relacional.

Outros modelos foram propostos para superar as limitações semânticas dos modelos orientados a registro, tais como: modelo de entidade relacionamento (MER) [Che76], o modelo funcional e o modelo de dados semântico [PM88].

Os sistemas de banco de dados mais utilizados pela indústria, atualmente, são os baseados no modelo relacional. Eles são fáceis de usar, têm boa base teórica mas mostram-se pobres para consultas e não comportam os modelos de dados e funcionalidades para aplicações avançadas que necessitam recursos gráficos e de multimídia como CAD (*computer aided design*) e SIG (sistemas de informações geográficas), por exemplo.

A nova tecnologia de SGBD orientada a objetos apresenta os recursos necessários para modelagem de objetos complexos e em constante evolução e aqueles conectados por relações semânticas complexas [Kim90]. Este modelo permite um mapeamento mais natural dos objetos do mundo real, melhorando o entendimento do problema e das consultas por parte do usuário, porque está mais voltado para os aspectos comportamentais dos objetos e não com aspectos estruturais. Isto pode ser comprovado através dos seus conceitos: encapsulamento, tipos e classes de objetos, hierarquia de tipos, hierarquia de classes e métodos. Esta nova tecnologia ainda não se firmou por ter alguns pontos para serem aprimorados como: otimização de consultas, segurança, tolerância a falhas, pelos altos investimentos em migração das aplicações e treinamentos necessários para a nova tecnologia.

A maioria das grandes organizações utiliza diferentes tipos de SGBD por inúmeras razões: históricas, econômicas e especificidade das aplicações. Se por um lado esta diversidade pode gerar aplicações mais específicas e que atendem a necessidade individual de cada parte das organizações, por outro lado, acaba dificultando as consultas e consequentemente a integração do dados localizados nos diferentes SGBD, devido principalmente à incompatibilidade dos modelos de dados e à falta de uma linguagem de acesso padronizada. Esta dificuldade na integração gera a redundância e replicação dos dados, dificultando a tomada de decisões, tornando as organizações estáticas em relação aos concorrentes, além de aumentar o custo de manutenção e diminuir a qualidade das aplicações existentes.

Inicialmente, tanto o SGBD quanto os dados por ele gerenciados estavam em um único computador. A arquitetura possuía de três níveis: nível interno que é a estrutura física dos dados, nível conceitual que é a estrutura lógica dos dados e nível externo que é a visão que cada usuário tem dos dados [TK78]

Com a revolução ocorrida nas áreas de redes de comunicação, começaram a aparecer novas maneiras de se construir SGBD, tornando-se possível separar as aplicações dos SGBD. Com isto pode-se aumentar a disponibilidade e o desempenho das aplicações, uma vez que os dados estariam espalhados pela rede, nos locais onde eles estariam sendo

	BD Distribuídos	Esquema Global de Múltiplos BD	BD Federados	Linguagem p/ Sistemas com Múltiplos BD	Linguagem p/ Sistemas com Múltiplos BD Homogêneos	Sistemas Interoperáveis
O sistema global tem acesso ...	às funções internas do SGBD local	à interface de usuário do SGBD local	à interface de usuário do SGBD local	à interface de usuário do SGBD local	à interface de usuário do SGBD local + funções internas	às aplicações sobre o SGBD local
Os nós locais são ...	BD homogêneos	BD heterogêneos	BD heterogêneos	BD heterogêneos	BD homogêneos	quaisquer tipos de dados
Possui funções globais completas de BD?	sim	sim	sim	sim	sim	não

Tabela 2.1: Grau de Acoplamento entre SBDH

mais acessados e também aproveitando o processamento distribuído, distribuindo a carga do sistema como um todo.

Para que os objetivos acima fossem alcançados, várias propriedades deveriam ser garantidas [Sto88]: transparência de localização, desempenho, replicação, transação e fragmentação. Estas propriedades são difíceis de serem alcançadas, por isso demorou-se tanto tempo para surgirem os primeiros SGBD comerciais que as implementassem, pelo menos em parte.

Além da distribuição é necessário que se construam sistemas que possam interoperar entre si, respeitando a autonomia de cada SBD componente (local) mas permitindo que os usuários possam operar sobre todos os dados, independentemente de qual SGBD está sendo responsável por controlar o dado. Várias nomenclaturas foram propostas para estes sistemas [Agu95] como: sistemas com múltiplos bancos de dados (*multidatabase systems*), sistema global, banco de dados federados. Nesta dissertação será usado o termo sistema de banco de dados heterogêneo (SBDH) por ser o nome mais genérico para referenciar um conjunto de bancos de dados com diferentes esquemas, protocolos de comunicação, *hardware* e linguagem de consulta.

A tabela 2.1 apresentada por Hurson et al. [HB91] mostra a variedade de SBDH e o grau de acoplamento de cada um.

Quanto mais acoplados forem os sistemas, mais difícil a administração e quanto menos acoplados menor o controle sobre as aplicações, menor o número de usuários capacitados para utilizar o sistema, isto é, capazes de construir visões sobre os SBD componentes e menor a possibilidade de se manter os BD locais íntegros em operações de atualização por uma possível interpretação semântica errada dos dados. Sendo assim, a utilização de sistemas fracamente acoplados são inadequados para aplicações tradicionais em banco de dados.

A diferença entre os sistemas de banco de dados distribuídos (SBDD) e os SBDH é

que no primeiro apesar dos dados estarem distribuídos fisicamente, o controle deles é feito por apenas um SGBD. Além disso os dados estão em um mesmo modelo de dados e sob os mesmos critérios de gerência de dados, transação e segurança. A diferença entre SBDH e sistemas interoperáveis é que este último não apresenta todas as funções de acesso de BD para usuários globais. Os SBDH apresentam duas características que dificultam o quadro da distribuição física dos dados: heterogeneidade e autonomia dos SBD componentes.

2.1.1 Heterogeneidade, Autonomia e Distribuição (HAD)

Como já descrito na seção anterior a distribuição dos dados é uma característica comum entre os SBDD e SBDH, portanto todos os problemas já existentes no contexto de SBDD devem ser levados em consideração nos SBDH. Sendo assim a transparência de localização, de desempenho, de replicação, de transação e de fragmentação devem ser respeitadas pelos SBDH mas com um grau de dificuldade muito maior porque deve-se garantir a autonomia dos SBD componentes que podem apresentar modelos de dados, controle de segurança, gerenciamento de transação e linguagem de consulta diferentes.

A heterogeneidade pode ser vista sob o aspecto dos SBD componentes em relação ao modelo de dados adotado, às restrições de integridade suportadas, à linguagem de manipulação, aos mecanismos de gerência de transação, aos sistemas operacionais, *hardware* e protocolos de comunicação. E sob o aspecto relacionado às diferenças semânticas dos dados nos vários SBD componentes devido às diferentes percepções dos projetistas sobre o mundo real, à variedade de construtores associados aos modelos de dados e à incapacidade destes de capturarem a semântica do mundo real.

O último ponto que caracteriza os SBDH é a autonomia dos SBD componentes. Segundo Sheth et al. [SL90] podemos ter diferentes níveis de autonomia:

1. projeto - cada SBD tem liberdade para definir a organização dos seus dados, linguagem de consulta, restrições implementadas além dos nomes e significados semânticos dos dados;
2. comunicação - capacidade de cada SGBD definir quando e como ele vai se comunicar com outros SGBD componentes;
3. associação - cada SBD componente decide quando e como participar de um SBDH;
4. execução - cada SBD componente tem a possibilidade de executar as transações sobre os seus dados sem nenhuma interferência ou controle externo.

Portanto os três aspectos principais que dificultam a integração de SBDH são: heterogeneidade, autonomia e distribuição. Eles devem ser resolvidos mesmo que relaxados em certos aspectos, principalmente autonomia, para obter SBDH eficientes e confiáveis.

Outros aspectos ligados à construção de SBDH são mecanismos de gerenciamento de transação globais que garantam a integridade dos dados em caso de falha em um dos SBD que compõem o SBDH. Hoje já existem algumas ferramentas comerciais que se propõem a fazer este serviço, os *TP monitors* (*transaction process monitors*). Outro aspecto são os mecanismos de segurança do SBDH, sem que este interfira na autonomia dos SBD locais.

2.1.2 Abordagens para Integração de SBDH

Existem inúmeras soluções para construir SBDH, principalmente sistemas de banco de dados federados (SBDF). Soares [Soa98] faz uma extensa revisão das abordagens para integração de SBDH. Possíveis soluções são: Mediadores [Wie92] [Ber96] [ULM97, LPO97], Visões [Cer96, Agu95, SP94], Metabanco de dados [HBRY91], Conversão de Esquemas e modelos de dados [Oli93, OMa93], API para acesso a SBDH [Kam96]. Esta dissertação irá focar o uso de mediadores¹ como solução para integração de SBDH.

A maioria das propostas de criação de SBDH utilizam a arquitetura de referência proposta por Sheth e Larson [SL90] (fig. 2.1) ou uma simplificação dela. A arquitetura leva em consideração as características de heterogeneidade, distribuição e autonomia dos SBD locais.

A transformação de modelos para um modelo de dados comum (MDC) e o processamento de consultas globais (PCG) também aparecem na maioria das soluções citadas para criação de SBDH.

Arquitetura de Referência

A arquitetura proposta por Sheth e Larson [SL90] é composta por transformações e filtragens (fig. 2.1), e é resumida a seguir.

O esquema local representa o esquema conceitual do SBD componente expresso no modelo de dados do seu SGBD local. O esquema componente representa o esquema local sobre o modelo de dados comum, uniformizando os conceitos modelados por cada esquema. Esta transformação também permite que operações sobre um esquema no MDC possam ser transformadas em operações no modelo de dados do SGBD local.

O esquema exportável é um subconjunto dos dados, autorizados para serem compartilhados, do esquema componente. Com esta filtragem consegue-se a autonomia de associação dos SBD componentes.

O esquema federado é formado por um processo que integra os esquemas exportáveis formando um esquema conceitual virtual a partir dos esquemas exportáveis. Serão geradas informações relativas à distribuição dos dados, permitindo que transações globais

¹mediador é um *software* que faz a integração entre componentes de dois ou mais sistemas que por algum tipo de restrição não conseguem trocar informações

envolvendo vários SBD componentes possam ser desmembradas em subtransações e posteriormente unidas.

Nos esquemas externos são formadas visões particulares dos dados nos esquemas federados para uma classe de usuários e/ou aplicações. Pode-se criar restrições de integridade e segurança adicionais que serão garantidas pelo processador de filtragem. Neste processo pode-se criar uma visão do esquema federado, construída segundo um modelo de dados diferente do MDC.

Modelo de Dados Comum

Para permitir que os usuários e aplicações atuem sobre os dados em SBD distintos de forma transparente, são utilizados os mapeamentos de esquemas locais em esquemas componentes e de esquemas federados em esquemas externos. Este mapeamento é feito em duas etapas: primeiro é feita a tradução do esquema em um modelo de dados para esquemas equivalentes em outro modelo de dados; depois é feito o mapeamento de operações expressa em uma linguagem de manipulação de dados utilizada em um modelo em operações da linguagem usada pelo outro modelo de dados (fig. 2.2).

Para fazer o mapeamento entre os vários modelos pode-se mapear diretamente cada par de modelos. Para N modelos teríamos $N * (N - 1)$ mapeamentos. A outra forma seria mapear todos os modelos para um modelo único, com uma linguagem associada a este para servir de interface. Neste caso teríamos $(2 * N)$ mapeamentos.

O modelo de dados comum (MDC) é de extrema importância para se obter transparência de modelos de dados e integração do SBD componentes em SBDH. Mas para isso, é necessário que ele tenha algumas propriedades:

1. facilidade para representar de forma simples e clara as informações contidas em outros modelos;
2. suportar diferentes níveis de abstração;
3. possibilidade de acrescentar informações adicionais em relação aos esquemas dos SBD componentes;
4. Apresentar uma linguagem de manipulação e definição de dados para ser utilizada como linguagem intermediária.

Vários projetos de pesquisa utilizaram o modelo relacional [Che76] e modelos semânticos [PM88] como modelos de dados comum. Atualmente o modelo de dados orientado a objetos [Agu95] é o modelo que tem sido mais utilizado para servir de modelo de dados comum [Kam96], por conseguir representar com maior facilidade os modelos de dados anteriores

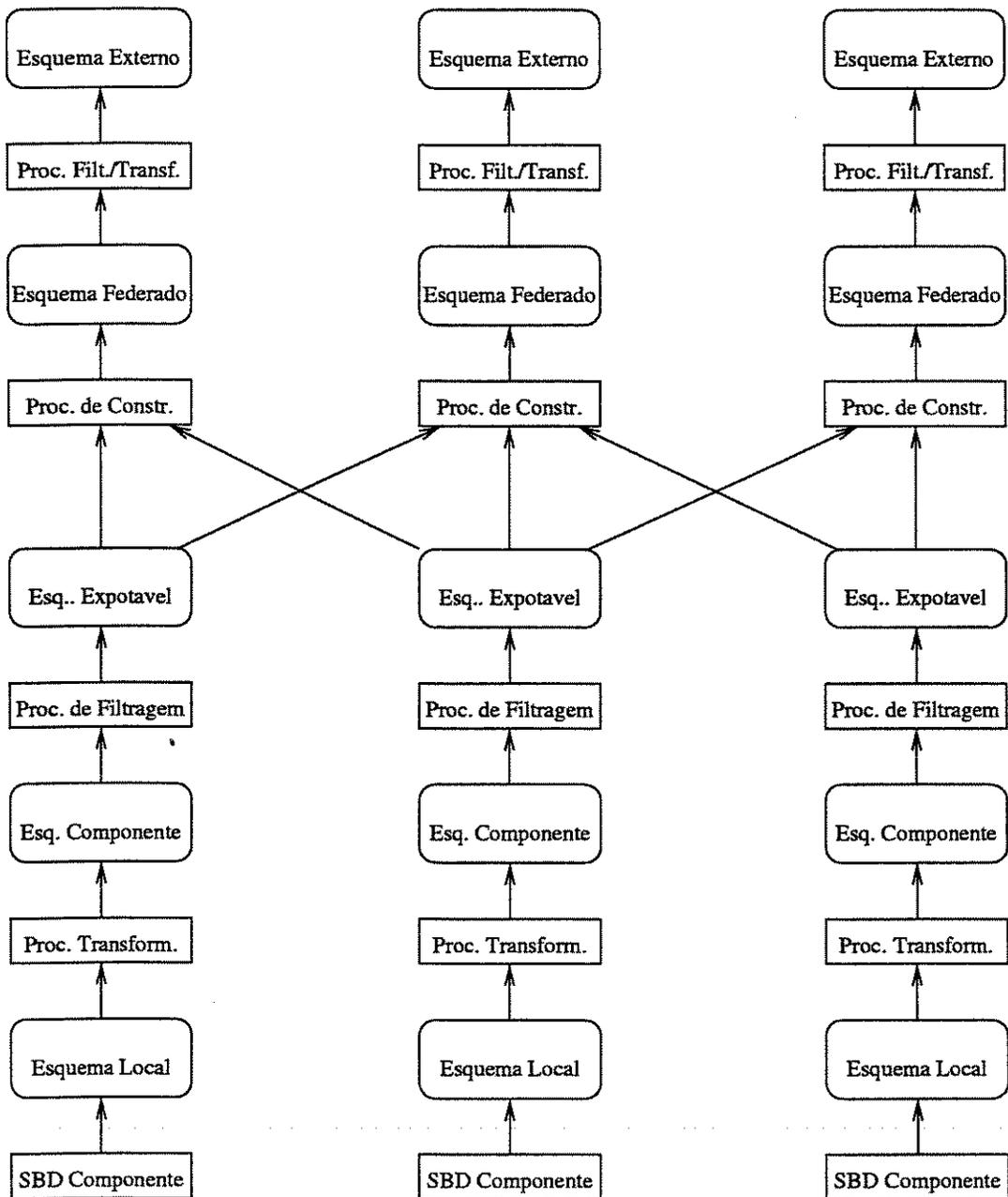


Figura 2.1: Arquitetura de Referência para SBDH [SL90]

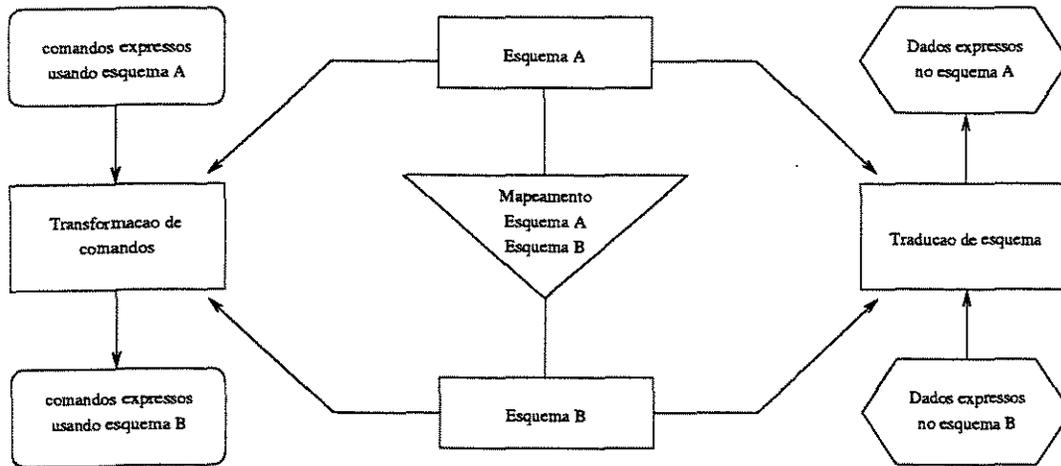


Figura 2.2: Transparência de Modelos

e também por conseguir integrar SBD convencionais e não convencionais como imagens, texto e outros.

Processamento de Consultas Globais

Uma visão simples do processamento de consultas utilizando-se linguagens de consulta global sobre um esquema global, pode ser visualizado através de três principais etapas (fig. 2.3):

1. decomposição em subcomandos que atuam sobre os dados de um único SGBD local;
2. cada subcomando é traduzido para o esquema local e executado pelo SGBD local;
3. combinação dos resultados locais para obter-se a resposta global (comandos de pós-processamento).

A decomposição da consulta global em subcomandos (etapas 1 e 2 acima) é reponsável pela maneira como os dados são transmitidos de um SGBD para outro e pelo processamento local a ser realizado, influenciando fortemente no desempenho da consulta global [Che90]. A decomposição pode ser vista em dois passos:

1. a consulta global contendo nomes globais é traduzida para comandos usando somente nomes dos esquemas dos SGBD locais mesmo que cada comando tenha dados de mais de um SGBD;

- esses comandos traduzidos são decompostos para poderem ser executados em um único SGBD local. Esses subcomandos podem apresentar aninhamento, isto é, um subcomando depende de dados de outro subcomando para ser executado.

Após a decomposição, cada subcomando ainda está expresso na linguagem de consultas global. Caso esta linguagem seja diferente da linguagem de consulta local, ela deve ser traduzida. Sendo assim diferentes tradutores entre linguagens são necessárias. Cada tradutor deve levar em consideração dois pontos:

- a correção da tradução;
- o desempenho dos comandos traduzidos.

Alguns comandos existentes na linguagem global podem não serem passíveis de tradução, mesmo que em vários comandos, para a linguagem local; neste caso o comando deve produzir um superconjunto como resposta de cada SGBD local e realizar um pós-processamento ao final de todas as consultas locais.

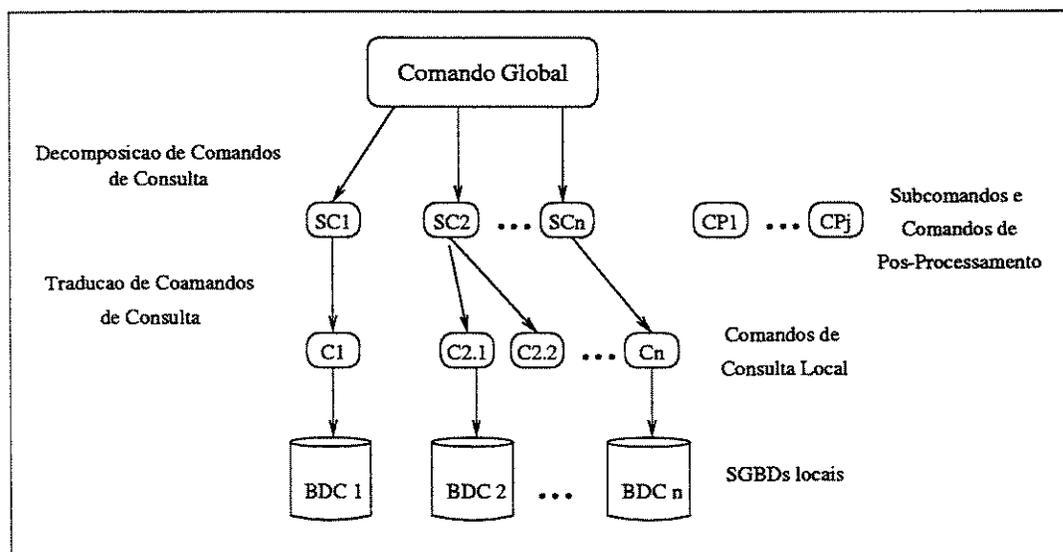


Figura 2.3: Processamento de Consultas Globais

Mediadores

Vários trabalhos têm utilizado mediadores para construção de SBDH: [VB97] [Wie92] [Ber96]. Segundo Wiederhold [Wie92] mediadores são módulos de *software* que exploram

o conhecimento representado em um conjunto ou subconjunto de dados para gerar informações residentes em uma camada superior. Já Soares [Soa98] é mais específico para a utilização de SBD, definindo mediador como um *software* usado para permitir a interoperabilidade entre dois ou mais SGBD. Nesta dissertação o termo mediador será utilizado de forma mais abrangente, como sendo um *software* que faz a integração entre componentes de *software* que por algum tipo de restrição não conseguem trocar informações. No trabalho de Brodie et. al [BS95] eles classificam mediadores em três categorias: Interface, Aplicação e Banco de Dados.

Seguindo as classificações de Brodie et al. [BS95] podemos classificar os *middleware services* como sendo um tipo especial de mediador de comunicação. Bernstein [Ber96] define *middleware services* como serviços que estão entre as plataformas² e as aplicações. Eles são genéricos para qualquer tipo de aplicação, podem ser executados em várias plataformas, são distribuídos e suportam interfaces e protocolos padrões. Os serviços são inúmeros, dentre eles: servidores de diretório, conversão e manipulação de *strings*, serviços de comunicação como mensagens ponto a ponto, chamada de procedimentos remoto, fila de mensagens e troca eletrônica de dados, serviços de segurança, serviços de controle de transação e *threads*, serviços de autenticação e criptografia.

O trabalho de Vidal et al. [VB97] apresenta uma metodologia para projeto de mediadores em três passos:

1. definição do esquema do mediador: definição dos requisitos do usuário e especificação de um modelo de dados de alto nível;
2. integração do esquema do mediador: integração do esquema do mediador com os esquemas locais para identificar as assertivas de correspondência entre o esquema do mediador e os esquemas locais;
3. especificação do mediador: definição do mapeador de estados e de atualização do mediador.

Vidal et al. [VB97] utilizam uma arquitetura de mediadores que é uma simplificação da arquitetura proposta por Sheth e Larson [SL90]. Para dar suporte à arquitetura, utilizam o modelo de entidade e relacionamento estendido para torná-lo mais flexível para expressar as várias formas de correspondência entre esquemas. Além do modelo, utilizam uma linguagem de consulta comum para permitir que novos mediadores sejam adicionados aos já existentes, aumentando o número de SGBD locais ao SBDH.

A figura 2.4 apresenta a arquitetura em três camadas proposta por Wiederhold [Wie92]. Nesta arquitetura existe o particionamento horizontal (usuários finais, mediadores e BD) e o particionamento vertical que são as várias aplicações dos usuários acessando várias

² *Hardware*, Sistema Operacional, Protocolo de Comunicação

configurações de mediadores que por sua vez irão utilizar visões distintas dos dados sobre um ou vários BD.

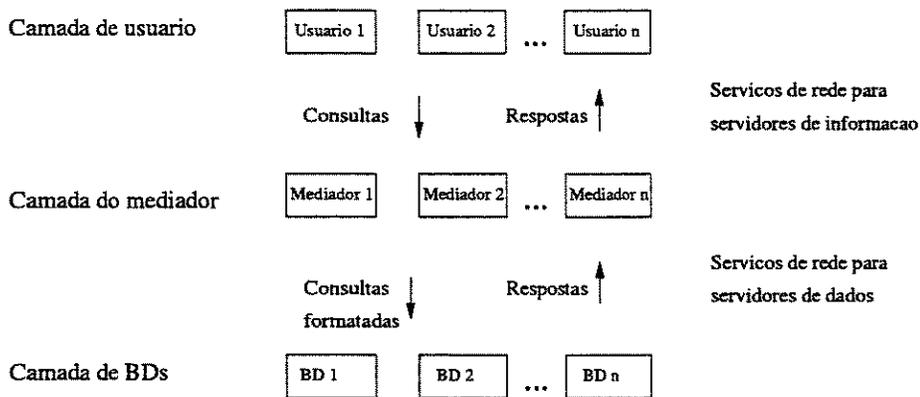


Figura 2.4: Arquitetura de Mediadores em três Camadas

Para a construção de mediadores utilizando esta arquitetura, os principais itens são [Wie92]:

- interfaces do mediador - as interfaces da camada do mediador são a parte mais importante na arquitetura de três camadas. Elas permitem que os usuários de qualquer sistema de informação tenha acesso de forma flexível aos dados armazenados em qualquer SGBD local. As interfaces entre os mediadores e os sistemas de informação pode ser conseguidas através de uma linguagem de alto nível. As interfaces entre os mediadores e os SGBD pode ser conseguidas através de padrões de banco de dados já existentes, como: SQL (*structured query language*) [MS92] e RDA (*remote data access*).
- compartilhamento dos módulos de mediadores - permite que os mediadores possam ser utilizados por uma grande variedade de aplicações;
- distribuição dos mediadores - os mediadores devem ser módulos de *software* e devem estar distribuídos pela rede, principalmente para manutenção e autonomia além de economia de acesso;
- gatilho para manutenção do conhecimento - gatilhos ou *daemons* permitem que mudanças nos BD sejam enviadas para os mediadores para que estes possam refletir, nas suas bases de conhecimento, as mudanças ocorridas no mundo real.

O trabalho apresentado por Uchôa et al. [ULM97] apresenta o SBDH HEROS (heterogeneous object system) em desenvolvimento na PUC-Rio e a integração do padrão

CORBA³ (*common object request broker architecture*) para a interoperabilidade dos componentes do sistema HEROS. O projeto HEROS é um SBDH fortemente acoplado que utiliza o modelo orientado a objetos como modelo de dados global. A arquitetura é a mesma proposta por Sheth e Larson [SL90], composta de esquemas locais, exportáveis, global e externos. A integração da arquitetura CORBA ao HEROS aumentou a flexibilidade e extensibilidade dos componentes do sistema uma vez que esta preconiza a interação de componentes heterogêneos de forma independente da sua implementação. Os componentes que foram alterados e tiveram maior benefício da arquitetura CORBA foram os componentes de gerência de transação distribuída e integração de esquemas.

Lima et al. [LPO97] apresentam a implementação de uma ferramenta de mapeamento de esquemas conceituais de SBDH utilizando a arquitetura CORBA. O mapeamento de esquemas para uma federação fracamente acoplada apresenta os seguintes passos:

1. geração do Esquema de Exportação (EE) a partir dos esquema local;
2. elaboração das Tabelas de Objetos Locais (TOL) que contenham informações referentes ao conjunto de objetos do esquema de exportação correspondente, a partir dos EE;
3. comparação dos EE e TOL resultando na criação de duas outras tabelas: Tabela de Equivalência Objetos (TEO) que e Tabela de Equivalência de Atributos (TEA).

A ferramenta para integração de esquemas possui o módulo servidor, responsável pelo gerenciamento das TEA, TEO, EE e TOL; e o módulo cliente, responsável pelo armazenamento das equivalências, detecção de conflitos e geração de interfaces HTML (*hypertext markup language*) para interação com o usuário.

2.2 Sistemas de Informações Legados

Atualmente os sistemas de informações legados são um problema dominante em várias organizações e grandes indústrias. Os maiores problemas enfrentados são a manutenção dos sistemas para que eles possam continuar atendendo as necessidades dos negócios e a outra é a dificuldade em deixá-los operacionais, devido principalmente à defasagem tecnológica. Apesar destas características, a tecnologia não deve ser o fator que leva qualquer organização a reestruturar o seu negócio. Os requisitos dos negócios é que devem levar a uma mudança na tecnologia e nos sistemas de informação.

Segundo Brodie et al. [BS95] um sistema de informação legado é qualquer sistema de informação que resiste à mudanças e evoluções em grande parte às dificuldades em

³Arquitetura padrão para interação de sistemas proposta pelo OMG

manutenção. Desta forma, não somente sistemas com com milhares ou milhões de linhas de código, escritos em COBOL mas também sistemas mais modernos, escritos em C ou C++, podem ser exemplos de sistemas legados. Outra característica dos sistemas legados é que eles costumam consumir grande parte dos recursos financeiros do total aplicado em sistemas de informação de uma organização, impedindo, muitas vezes, a sua migração para que possam atender as necessidades dos negócios.

O desenvolvimento de sistemas centralizados, baseados em mainframes, modelos de desenvolvimento totalmente dependente do *hardware* e *software* utilizado, interfaces de usuário baseadas em terminais não gráficos e processamento em *batch*, predominante nas últimas décadas, levou várias empresas a uma verdadeira crise pois suas aplicações não conseguem evoluir com a mesma rapidez que os negócios exigem. Além disso, elas são totalmente dependentes deste tipo de sistemas, o que os tornam sistemas de missão crítica [PMa02] para estas empresas.

As características dos sistemas legados são [BS95]:

1. são grandes, com milhões de linhas de código;
2. eles são muito velhos, mais de dez anos e passaram por várias reestruturações;
3. usam linguagem legadas como COBOL;
4. foram construídos sobre um sistema de banco de dados legado, por exemplo, IMS ou sobre estruturas de arquivos como: ISAM e VSAM;
5. são autônomos, isto é, não têm interface com nenhuma outra aplicação;
6. apresentam baixo desempenho;
7. possuem pouca ou nenhuma documentação e geralmente estão desatualizadas.

Os sistemas de informações legados tradicionais são aqueles baseados em mainframe porém o paradigma de sistemas legados pode ser aplicado a qualquer plataforma. Características como aumento dos custos de manutenção, aumento do intervalo de tempo em incorporar novas funcionalidades podem representar um sistema de informações legado em formação.

Apesar deste tópico estar sendo pesquisado há anos e ser um assunto muito discutido e exposto, além de existirem várias ferramentas que se propõem a traduzir estes sistemas, pouco tem sido feito neste sentido pelas grandes organizações e quando existe este esforço, o sistema é totalmente refeito. Para que o processo de migração dos sistemas legados para uma nova tecnologia tenha sucesso, é necessário ter um plano ou estratégia para esta migração, uma vez que os sistemas legados não podem parar, isto é, continuam em evolução, enquanto a migração está acontecendo.

Brodie et al. [BS95] apresentam duas estratégias de migração: *Cold Turkey* e *Chicken Little*

2.2.1 Estratégia *Cold Turkey*

Esta estratégia preconiza a migração de todo o sistema legado de uma só vez: todo o sistema deve ser analisado utilizando, por exemplo, técnicas de engenharia reversa. O sistema é integralmente refeito e os seus dados são convertidos para um SGBD. Isto deve ser feito com ajuda de ferramentas de especificação e geração de código e interface humano-computador.

Para Brodie et al. [BS95] esta estratégia apresenta vários problemas com grande possibilidade de falhar:

1. um sistema melhor deve ser prometido para o usuário;
2. as regras de negócio da empresa nunca são estáticas;
3. especificações do sistema legado raramente existem;
4. sistemas legados podem ser muito grandes para que haja uma migração dos dados de uma só vez;
5. o gerenciamento de grandes projetos é uma tarefa difícil;
6. atrasos são pouco tolerados;
7. grandes projetos tendem a inchar com o tempo;
8. o medo de mudanças é predominante;
9. a análise do sistema como um todo pode levar a uma paralisia.

2.2.2 Estratégia *Chicken Little*

Esta estratégia é bem mais conservadora e propõe-se a fazer uma migração incremental do sistema legado, parte a parte, até que o sistema como um todo esteja migrado. Isto irá permitir que a organização continue operando, atendendo seus clientes e as alterações vão sendo sentidas aos poucos. Os riscos são bem menores, uma vez que uma falha representa apenas reiniciar a migração da parte que está sendo migrada, e não a falha de todo o processo. Cada problema existente na estratégia *Cold Turkey* pode ser atacado agora de maneira incremental e em um escopo reduzido.

A estratégia *Chicken Little* de migração é composta por alguns elementos de migração individuais - os passos da migração, que cuidam de um aspecto específico da migração. Quanto maior a independência entre os passos, maior a flexibilidade para adaptar o método aos requisitos específicos de cada sistema, para absorver mudanças de requisitos ao longo da migração e para corrigir erros.

Os passos da migração são:

1. Analisar o sistema de informação legado incrementalmente;
2. Decompor a estrutura do sistema de informação legado incrementalmente;
3. Projetar as interfaces do sistema destino incrementalmente;
4. Projetar as aplicações destino incrementalmente;
5. Projetar o banco de dados destino incrementalmente;
6. Instalar o ambiente destino incrementalmente;
7. Criar e instalar os *gateways* necessários incrementalmente;
8. Migrar o banco de dados legado incrementalmente;
9. Migrar a aplicação legada incrementalmente;
10. Migrar as interfaces legadas incrementalmente;
11. Substituir o sistema legado incrementalmente.

Dependendo das características de cada sistema a ordem dos passos pode mudar ou certos passos podem ser unidos em um só passo. O grafo da figura 2.5 diz a ordem de execução dos passos. Os passos que estão no mesmo nível podem ser executados simultaneamente.

2.2.3 Trabalhos Correlatos

Para Ganti et al. [GB95] a transição dos sistemas legados pode ser de duas maneiras: *greenfield* - a migração será feita e o sistema legado será abandonado e *brownfield* - sistemas complexos não poderão ser simplesmente descartados, eles serão migrados gradualmente e integrados com os novos sistemas.

Uma importante consideração é não causar nenhum tipo de dano às funções de negócios que estão em produção, durante o processo de transição. Um bom entendimento destes

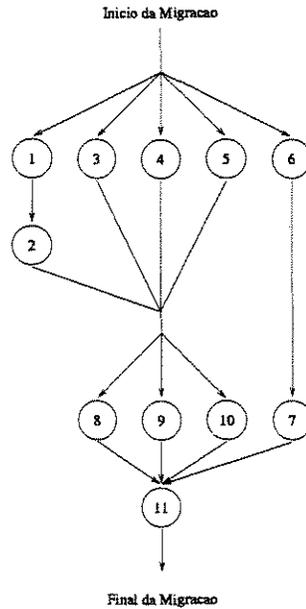


Figura 2.5: Grafo de Dependências entre os Passos da Migração

dois ambientes (velho e novo) é essencial para determinar o processo de transição. Portanto, é necessário desenvolver um processo cuidadoso de transição. O ponto inicial é definir os processos de negócio.

Alguns passos chave para desenvolver o processo de transição na metodologia de Ganti et al. [GB95] são:

1. A partir dos processos de negócio, determinar os dados necessários e fazer uma classificação lógica dos mesmos;
2. Classificar os dados de acordo com a sua propriedade (usuários e sistemas) e autonomia (privado ou compartilhado);
3. Criar classes de metadados baseadas nas regras dos processos de negócios;
4. Criar os bancos de dados de operação e de suporte à decisão baseado nos assuntos que eles tratam;
5. Desacoplar os procedimentos ou regras de negócios das aplicações e armazená-las como metadados ou *stored procedures* reutilizáveis;
6. Criar banco de dados de metadados para cada classe dos processos de negócios. Por exemplo: desenvolvimento de aplicações, dados de usuários que são apenas de

- leitura, desenvolvimento de aplicações para usuários finais;
7. Criar os bancos de dados de transação;
 8. Criar os bancos de dados de suporte à decisão;
 9. Definir replicação de dados e processos de sincronização para os armazéns de dados para suporte à decisão;
 10. Iniciar o plano de desenvolvimento e implantação da aplicação;
 11. Estabelecer processos para operar os sistemas legados em paralelo com novos sistemas distribuídos com transações distribuídas entre os dois sistemas (velho e novo), até que os sistemas legados sejam completamente migrados;
 12. Iniciar plano de implantação tecnológica;
 13. Iniciar o treinamento dos usuários;

Soares [Soa98] apresenta uma metodologia de integração de sistemas legados baseado em uma federação fracamente acoplada composta por bancos de dados heterogêneos. Soares concentra-se na parte dos dados legados e não nos aspectos da aplicação. A metodologia é composta de três grandes fases: padronização, exportação e criação do SBDF. Durante a padronização os diferentes modelos de dados dos SBD componentes da federação e os dados dos sistemas legados, quando possível, são traduzidos para um modelo de dados comum (MDC). Soares escolheu o modelo relacional [Cod70] acrescido de informações semânticas para conseguir mapear as diferentes semânticas dos diversos modelos de dados que compõem a federação. Na fase de exportação são definidos os dados de cada esquema local que farão parte de federação. Segundo Soares esta fase é dependente das aplicações que irão ser executadas sobre a federação, sendo o seu tratamento algorítmico bastante reduzido. Por fim para a fase de criação do SBDF, Soares utilizou o algoritmo apresentado por Zhao [Zha97] estendido pois este apresentava algumas deficiências quanto a chaves estrangeiras compostas, utilização de mais de um SBD componente em uma consulta, informações relativas aos domínios dos atributos e ambiguidade de caminhos para se obter a resposta a uma consulta.

A metodologia é dinâmica, isto é, ela preconiza que as fases podem realimentar fases anteriores, criando *ciclos de realimentação* e a mesma fazia parte do projeto SIGGER - sistema de informações gerenciais geograficamente referenciados, da prefeitura da cidade de Paulínia.

Outra importante contribuição do trabalho de Soares foi inserir os sistemas de informações geográficos na metodologia, levantando os possíveis problemas de diferenças de escala, representação de dados, projeção e aspectos temporais dos dados.

Markosian [MNB⁺94] descreve uma ferramenta para apoiar a análise e modificação sistemática dos sistemas legados existentes. Ele apresenta os resultados do beta-teste aplicado em um sistema de folha de pagamento da Boeing escrito em COBOL com aproximadamente 650.000 LOC (*Lines Of Code*). Já havia uma equipe de vinte pessoas trabalhando na reengenharia deste sistema, executando as seguintes atividades: modularização, substituição dos bancos de dados baseados em arquivos (*flat files*) por um SGBD relacional, extração das regras de negócio e reorganização da camada de aplicação (*procedure division* em COBOL) para as regras de negócio correspondentes.

A ferramenta se propõe apenas a automatizar a tarefa de modularização. A idéia central da ferramenta é representar o *software* em uma forma de árvores sintáticas abstratas (ASA) em um banco de dados orientado a objetos (BDOO) e analisar e transformar o código em uma linguagem de especificação de alto nível. Ela é composta por quatro módulos:

1. DIALECT - compilador gramatical que gera a especificação gramatical do sistema (estruturas de dados orientadas a objeto que representam as ASA);
2. REFINE - este módulo possui um BDOO utilizado para modelar o sistema e guardar informações sobre ele, uma linguagem executável de alto nível, um compilador e depurador para a linguagem e um biblioteca *run time* independente de linguagem para manipular as ASA;
3. WORKBENCH - uma biblioteca de componentes reusáveis para construção de ferramentas de reengenharia independentes de linguagem;
4. INTERVISTA - um *toolkit* para geração de interfaces gráficas baseadas no padrão *X-Windows*.

A utilização da ferramenta aumentou o desempenho da reengenharia e os primeiros testes indicaram que o código foi corretamente gerado o que não aconteceu com o código gerado artesanalmente.

Ning et al. [NEK94] também apresentam uma ferramenta de reengenharia para auxiliar no processo de análise e entendimento do código legado para posterior modularização: COBOL/SRE (*Cobol system renovation environment*). Ela é baseada no conceito RCR (*reusable component recovery*). Os principais componentes desta ferramenta são: *Workspace management*, *Data model recovery* e *Concept recognition*. Assim como Brodie [BS95] eles apresentam uma solução de migração paulatina do sistema legado, mantendo um sistema híbrido durante a migração.

A proposta de Robertson [Rob97] apresenta uma arquitetura de migração de sistemas legados baseada em programação dinâmica orientada a objetos, reflexão e linguagens de

domínio específico. Sua arquitetura é baseada em cinco componentes de *software* interativos: *Data Pump*, *Data Converter*, *Data mappings* (camada reflexiva), um *Browser* e uma Enciclopédia. Sua proposta é isolar as constantes mudanças no mapeamento dos dados entre os sistemas legados e os novos sistemas cliente/servidor através de uma camada de reflexão.

Aiken [AAR94] propõe a migração dos sistemas legados através de técnicas de engenharia reversa. Eles criaram uma metodologia de engenharia reversa para ser aplicada em diversas migrações de sistemas legados do DoD (*U.S. Department of Defense*). Os princípios básicos da metodologia são:

- modelo de dados;
- transformação de modelos;
- projeto físico e esquemas;
- BD e desenvolvimento de aplicações;
- plano de migração de dados e sistemas;
- plano de teste de dados e sistemas;
- plano de implementação de dados e sistemas.

Também geraram uma arquitetura de dados para guiar a migração de todos os sistemas de informação e para os novos desenvolvimentos.

2.3 Mapeamento dos Processos de Negócio

Na maioria dos trabalhos existentes na literatura sobre migração de sistemas legados, o enfoque maior está nos aspectos técnicos da migração como integração de sistemas, integração de banco de dados e conversão de dados. Existem poucos trabalhos que tratam dos requisitos que estão levando os sistemas a se tornarem legados. Os sistemas não se tornam legados apenas porque são antigos ou foram escritos em uma linguagem não mais utilizada ou executam em uma plataforma antiga. Normalmente os sistemas se tornam legados principalmente pela impossibilidade de serem alterados para atender os novos requisitos de negócio. Sendo assim antes de iniciar a transição dos sistemas legados para os novos sistemas, os requisitos do negócios devem ser analisados e os processos de negócio da organização que estão requisitando as mudanças devem ser mapeados para que o novo sistema de informação possam atendê-los.

Processos de negócio são as atividades internas desenvolvidas para atender os clientes. Quando o processo está dividido entre várias partes da organização, as interfaces devem estar muito bem definidas para que não haja conflitos organizacionais e pessoais.

Reengenharia dos processos do negócio implica em ter uma visão compreensiva e completa do negócio atual e mapear *o que, como e por que* ele é feito da forma atual. Assim todo o negócio é avaliado e deve ser pensado em uma nova maneira de se reconstruí-lo.

Davenport [Dav93] aponta a tecnologia da informação como sendo a facilitadora para a reengenharia dos processos do negócio por ela ser a área da empresa que integra as demais áreas. Um bom sistema de informação pode unir atividades diferentes dentro de um processo e pode fazer com que o fluxo de processos complexos se torne mais fácil uma vez que eles estarão sendo utilizados por diferentes áreas da empresa. Porém um dos riscos mais altos na reengenharia do processos do negócio está associado à construção do sistema de informação que irá suportá-los.

Para Jacobson et al. [JEJ95] um modelo de negócio mostra o que é o ambiente da companhia e como a companhia age em relação a seu ambiente. Ambiente são todas as entidades com quem a companhia interage para executar seus processos de negócio, como por exemplo: clientes, parceiros, subcontratados, fornecedores, etc. Ele deve mostrar aos empregados *o que* deve ser feito, *quando* e *como* deve ser feito dentro de cada área de empresa. A empresa não precisa ter apenas um modelo, ela pode ter vários modelos, um para cada área ou parte da empresa e eles devem estar integrados consistentemente.

O desenvolvimento dos modelos de negócio e os sistemas de informação devem ser feitos interativamente. Para desenvolver o sistema de informação devem ser acessadas várias informações sobre o negócio da empresa, como:

- Os usuários dentro e fora da companhia que utilizarão o sistema;
- Todos os processos de negócio e como cada um deve ser suportado pelo sistema de informação;
- Os tipos de documentos que são produzidos enquanto um processo de negócio é executado e o que deve ser feito com cada documento.

Para Jacobson et al. [JEJ95] o trabalho de descrever um negócio para poder reprojotá-lo consiste em duas fases:

- *reverse engineering* - entender a atual situação da empresa;
- *forward engineering* - projetar a nova companhia.

Nas duas situações é necessário saber *o que* o negócio deve fazer e para isto deve-se utilizar o modelo de caso de uso [FS97]. Para entender como o negócio irá funcionar, isto

é, os fluxos de dados e tarefas envolvidas, deve-se construir um ou mais modelos de objetos [FS97].

O modelo de caso de uso é composto por atores e casos de uso. Um ator representa uma entidade que interage com o sistema enquanto o caso de uso especifica um fluxo de eventos que um ator executa no negócio. O modelo de caso de uso auxilia a descrever o modelo da forma como os clientes gostariam de vê-lo. Ele deve ser entendido pelos clientes portanto não deve ser muito detalhado.

Após a descrição dos casos de uso, deve-se fazer um modelo de objetos do negócio. Para aumentar a expressividade do modelo de objetos deve-se utilizar três tipos de objetos: interface, controle e entidade. Os objetos de interface e controle geralmente representam tarefas do negócio enquanto que os objetos de entidade representam produtos ou qualquer outra coisa que é utilizada no fluxo dos eventos.

Sharon et al. [SB96] propõem uma metodologia de reengenharia do *workflow* da organização em cinco fases e também propõe a utilização de um sistema de gerenciamento de *workflow* para facilitar a reengenharia dos processos de negócio.

Na primeira fase é reconhecida a necessidade de uma melhoria na maneira como os processos de negócio são feitos. As mudanças na estruturas de liderança são estabelecidas e uma equipe de reengenharia é formada e treinada. Depois uma ferramenta de gerenciamento de *workflow* é procurada, comprada e instalada. Os ciclos de negócio são identificados e uma visão para o futuro da organização é estabelecido. O ambiente de projeto é definido e um dos ciclos do negócio é selecionado para ser melhorado.

Na segunda fase são identificados os produtos do negócio do ciclo de negócio que está sobre revisão e um único processo é selecionado para ser melhorado. O *workflow* deste processo é modelado e pode ser inserido em alguma ferramenta. As medidas de desempenho deste processo são monitoradas e armazenadas para a fase de melhoria do processo.

Após o *workflow* do processo estar automatizado e o seu desempenho estar sendo medido e armazenado, inicia-se a terceira fase. São identificadas as necessidades dos consumidores externos que utilizam o processo e este é avaliado para ser comparado com os concorrentes. Estes dados vão servir para definir os objetivos de desempenho para cada medida do processo. A partir deste ponto devem ser utilizados os princípios de reengenharia de processos de negócio como o trabalho desenvolvido por Hammer et al. [HC93]. Novos modelos de *workflow* são construídos e cada um é simulado para determinar suas características. Finalmente o modelo mais eficiente e efetivo deve ser escolhido.

Na quarta fase, as modificações na infraestrutura do *workflow* atual é especificada e instituída.

Na última fase o novo *workflow* é implantado e qualquer mudança na sua configuração do é sempre documentada. Uma atividade de melhoria contínua do processo é estabelecida

para garantir que o processo presente será periodicamente realinhado aos objetivos do negócio.

Seidmann et al. [SS96] propõem três classes de fatores para o realinhamento dos processos de negócio. A primeira consiste na natureza das tarefas que compõem o trabalho e o alinhamento operacional do sistema de trabalho. O segundo se refere aos aspectos dos recursos humanos do processo: compensação, análise de desempenho e poder de decisão. O terceiro se refere aos sistemas de informação e o suporte tecnológico para o processo.

Os sistemas de informação analisados por Seidmann et al. [SS96] pertencem a três tipos de tecnologias:

- tecnologia que aumenta a produtividade - sistemas que possam auxiliar na tomada de decisões, agilizar o fluxo e consulta de informações;
- expansão do escopo do processo - sistemas que expandem as qualidades dos trabalhadores e possibilita aos trabalhadores aumentar o escopo das tarefas que eles podem executar. Exemplo de sistemas são sistemas especialistas (*expert systems*) e sistemas de apoio à decisão;
- compartilhamento de informações - sistemas que permitem o acesso de informações *cross-functional* consolidadas de diferentes entidades da organização e permite compartilhamento de informações entre grupos de trabalhos. Exemplos de sistemas são sistemas de *groupware* ou sistemas baseados na *internet*.

Para Sundararajan [Sun96] o realinhamento dos processos é mais desejável quando existe um grande número de tarefas em um processo, alta variabilidade no tempo das tarefas, assimetria de informações e uma necessidade grande de personalização. Por outro lado o realinhamento dos processos é menos desejável quando os processos são intensamente baseados no conhecimento (*knowledge intensive*), as tarefas são poucas e uniformes e existe uma assimetria de informações ou baixo retorno no compartilhamento das informações.

Bracchi et al. [BF96] propõem que a informação possa ser compartilhada e adaptável. Compartilhamento refere-se ao nível que os agentes de informação trocam informações entre si. O compartilhamento é alto quando a quantidade de informação utilizada pelas tarefas é praticamente igual a quantidade de informações de entrada do processo. Adaptabilidade refere-se ao grau de filtragem das informações para que elas possam atender os requisitos individuais de cada tarefa. Quando as tarefas apresentam um alto grau de utilização das informações recebidas como entrada, elas estão recebendo informações ajustadas para os seus requisitos.

A tabela 2.2 apresentada por Bracchi et al. [BF96] classifica os processos de acordo com a sua interdependência e complexidade, uma vez que estes fatores estão diretamente relacionados ao compartilhamento e adaptabilidade das informações.

	Tarefas de baixa Complexidade	Tarefas de alta Complexidade
Tarefas de baixa Interdependência	Centralizadas sem banco de dados	Descentralizadas
Tarefas de alta Interdependência	Centralizadas com banco de dados	Distribuídas

Tabela 2.2: Classificação de Processos

Nas arquiteturas centralizadas sem banco de dados não há distinção entre dados e aplicações. Dados comuns a várias aplicações estão normalmente duplicados, pois as aplicações não conseguem se comunicar. Arquiteturas centralizadas que utilizam banco de dados permitem o projeto dos dados e aplicações separadamente. Os dados comuns a várias aplicações podem ser integrados e gerenciados unificadamente. Nos dois casos os dados e processamentos são centralizados.

Arquiteturas descentralizadas são caracterizadas pela capacidade de computação local ao *site* dos usuários. Já as arquiteturas distribuídas são marcadas pela comunicação entre os diversos *sites* e o armazenamento dos dados e execução das aplicações pode ocorrer em qualquer lugar. A maior diferença entre as duas arquiteturas está na forma de armazenagem dos dados, enquanto que na arquitetura distribuída os dados não estão duplicados e devem ser replicados para possibilitar o acesso em vários *sites*, na arquitetura distribuída os dados podem ser acessados de qualquer *site* da arquitetura.

Nos casos reais as arquiteturas são híbridas, isto é, uma mistura das várias arquiteturas citadas acima, principalmente devido à presença de sistemas legados dentro da organização.

Bracchi et al. [BF96] avaliam ainda o custo de comunicação, processamento, armazenagem e sobrecarga de cada uma das arquiteturas. A tabela 2.3 apresenta esta avaliação.

Para Bergey et al. [BNS97] as organizações que fazem a migração dos sistemas legados para sistemas distribuídos e abertos na maioria das vezes falham por se concentrarem em um pequeno conjunto de questões técnicas sem considerar todo o conjunto de engenharia de sistemas, a crescente necessidade dos clientes, o planejamento estratégico dos negócios, os objetivos da organização e a operação dos negócios da empresa. Eles propõem um *framework* para a organização, composto de sete elementos que são os blocos de construção para reengenharia ou evolução bem sucedida dos sistemas. Os elementos são: organização, projeto, sistemas legados, engenharia de sistemas, engenharia de *software*, tecnologias e sistema alvo.

Para cada elemento, Bergey et al. [BNS97] provêem uma lista de pontos a serem

	Custo de Comunicação	Custo de Processamento	Custo de Armazenamento	Custo de Overhead
Centralizadas sem banco de dados	++	++	+++	+
Centralizadas com banco de dados	+	++	++	++
Descentralizadas	++++	+	+	+
Distribuídas	+++	+	+	+++

Tabela 2.3: Avaliação de Arquiteturas quanto a Distribuição

verificados para guiar a análise do elemento durante a evolução do sistema. Esta lista serve para identificar os tomadores de decisão e quais fatores guiam as atividades e processos de tomada de decisão. Estas listas também servem para identificar zonas-cinza no planejamento da empresa como por exemplo: como aspectos interdependentes do trabalho serão coordenados com organizações externas e clientes e como os sistemas propostos serão potencialmente afetados por outros esforços de migração em planejamento ou em execução.

2.4 Arquitetura OMG

Com a crescente diminuição dos custos do *hardware* e o aumento dos custos e necessidades de *software* além da falta de qualidade e cooperação, levou um grande número de fabricantes de produtos de *hardware* e *software* de todo o mundo a se reunirem fundando, em 1989, o *Object Management Group* (OMG) [Gro93]. Seu objetivo é adotar uma arquitetura padrão, detalhando as especificações das interfaces, para possibilitar a interoperação, portabilidade e reusabilidade entre softwares - especificamente, os orientados a objetos - de diferentes tecnologias e de diferentes fornecedores. Nesta seção será apresentada a arquitetura do OMG, OMA (*Object Management Architecture*), uma especificação de arquitetura e interface que permite aplicações fazerem pedidos de operações sobre objetos de maneira transparente e independente de implementação e localização. A implementação destes objetos (dados e métodos) é responsabilidade dos vendedores e usuários finais. O OMG preocupa-se apenas com a especificação da interface destes componentes. Desta forma, é garantido que os produtos que resolverem adotar o padrão estabelecido pelo OMG poderão intercomunicar-se, uma vez que terão a mesma interface [WSM01]. A seguir

será apresentado o Modelo de Objetos OMG onde serão apontadas suas características principais bem como seus objetivos. Em seguida será mostrada a arquitetura do OMG.

2.4.1 O Modelo de Objetos do OMG

A idéia inicial da tecnologia Orientada a Objetos é a de facilitar a interoperação entre diferentes plataformas de *hardware* e *software*. Porém ao, contrário do que se esperava esta tecnologia está contribuindo para que vários modelos de objetos apareçam, cada um com o seu modelo de interação. Por esta razão o OMG decidiu criar um núcleo de um Modelo de Objetos que fosse comum a todas as aplicações que desejam adotar os padrões do OMG. Este núcleo é composto por um pequeno número de conceitos básicos. Primeiro os conceitos significativos para os clientes⁴, isto é, a semântica dos objetos: objetos, operações, tipos e subtipos, interfaces, atributos, pedidos⁵ e assinatura⁶. E depois os conceitos de implementações de objetos [MM97]. O modelo de objetos é mais específico e direcionado para a definição dos conceitos significativos para os clientes. A implementação dos objetos é muito mais sugestiva, permitindo assim uma grande liberdade para diferentes tecnologias de objetos proverem maneiras diferentes de implementação de objetos.

Os componentes da semântica dos Objetos serão apresentados a seguir.

Objetos

Objetos podem representar qualquer tipo de entidade do mundo real. Sua principal característica é sua identidade, isto é, a referência do objeto. Ela é imutável enquanto o objeto existir, ao passo que outras características podem variar ao longo do tempo. Um objeto pode ser referenciado por múltiplas e distintas referências. Uma referência de um objeto é o resultado de um pedido por um cliente. A implementação da referência dos objetos não faz parte do escopo do Núcleo do Modelo de Objetos.

Atributos

São entidades que podem ser definidas logicamente equivalente a declaração de duas funções, uma para recuperar o valor do atributo e outra para gravar o seu valor.

Operações

Uma operação é uma entidade identificável que representa um serviço que pode ser requisitado. Ela descreve uma ação que pode ser aplicada sobre parâmetros. Operações

⁴cliente é qualquer entidade capaz de requisitar um serviço [Gro93]

⁵pedido é um evento disparado por um cliente quando este requisita um serviço

⁶assinatura é a descrição dos parâmetros que compõem um pedido

possuem um identificador e têm uma assinatura que consiste em:

- especificação dos parâmetros necessários no pedido da operação;
- especificação dos resultados;
- especificação das exceções que podem acontecer e os tipos dos parâmetros que as acompanha;
- especificação de informação contextual adicional que pode afetar o pedido;
- indicação da semântica de execução que o cliente espera do pedido de uma operação.

Tipos

Um tipo é uma entidade com um predicado associado definido sobre valores. Um valor é parte de um tipo se o predicado é verdadeiro para esse valor. Objetos são criados como instâncias de tipos. Um tipo caracteriza o comportamento de suas instâncias descrevendo as operações que podem ser aplicadas sobre elas. A maneira como estes tipos devem ser implementados não é abordado pelo modelo de objetos. É assumido apenas que estas implementações existem e que para um mesmo tipo podem existir várias implementações. Tipos participam de relacionamentos de subtipo e supertipo os quais afetam o conjunto de operações aplicáveis às suas instâncias. Para cada operação de um supertipo existe uma operação correspondente para o seu subtipo que satisfaz as condições:

- Os nomes das operações são iguais;
- O número e tipo dos parâmetros são os mesmos com exceção do parâmetro de controle⁷;
- O número e tipo dos resultados são os mesmos.

Um tipo pode ter vários subtipos no modelo de objetos, porém as condições acima não permitem que dois tipos tenham operações com mesmo nome e diferentes assinaturas. Herança é o relacionamento de subtipo e supertipo aplicado sobre as operações e implementações dos objetos. Porém o núcleo do modelo de objetos não especifica o que pode acontecer com a implementação de operações herdadas. O modelo suporta herança múltipla a qual permite que um subtipo tenha múltiplos subtipos.

⁷ parâmetro distinto em relação aos demais e que contém o resultado da operação

Interface

Um tipo exporta todas as suas operações que são definidas sobre ele, caracterizando assim a sua interface. Um objeto satisfaz uma interface se ele pode ser especificado como o objeto destino de cada possível pedido descrito pela interface. O conjunto de assinaturas das operações de um objeto, isto é, sua interface, define o seu comportamento [OMG98b].

Como já notado, o núcleo do Modelo de Objetos especifica apenas a semântica dos objetos, dizendo pouco sobre sua implementação. A implementação de um sistema de objetos executa as atividades computacionais necessárias para afetar o comportamento dos serviços pedidos. Essas atividades podem incluir a computação do resultado de um pedido e a atualização do estado⁸ do sistema. A Implementação de Objetos pode ser dividida didaticamente nos modelos abaixo:

Modelo de Execução

O código necessário para a executar um serviço é chamado método. Um método é executado por uma máquina de execução que é uma máquina abstrata que pode interpretar métodos de um determinado formato, executando a computação associada ao método (ativação do método). Quando um cliente faz um pedido, um método do objeto destino é chamado. Se a sua forma persistente não está disponível para a máquina de execução, será necessário fazer uma cópia do método para o contexto de execução, este mecanismo é chamado ativação e o mecanismo contrário é chamado desativação [OMG98a].

Modelo de Construção

Para executar um método são necessários mecanismos para isto, tais como:

- definição do estado do objeto;
- como selecionar o método para ser executado;
- como selecionar o estado do objeto relevante para a execução do método;
- ações a serem executadas para a criação do método.

2.4.2 OMA

O objetivo da arquitetura do OMG é identificar e especificar os componentes, interfaces e protocolos para que as aplicações possam ser construídas de maneira interoperável e de maneira distribuída para todos os *hardwares*, sistemas operacionais e linguagens de

⁸estado é a informação necessária para alterar o resultado de operações subsequentes

programação. A figura 2.6 mostra os componentes da OMA e seu relacionamento via ORB⁹ (*object request broker*).

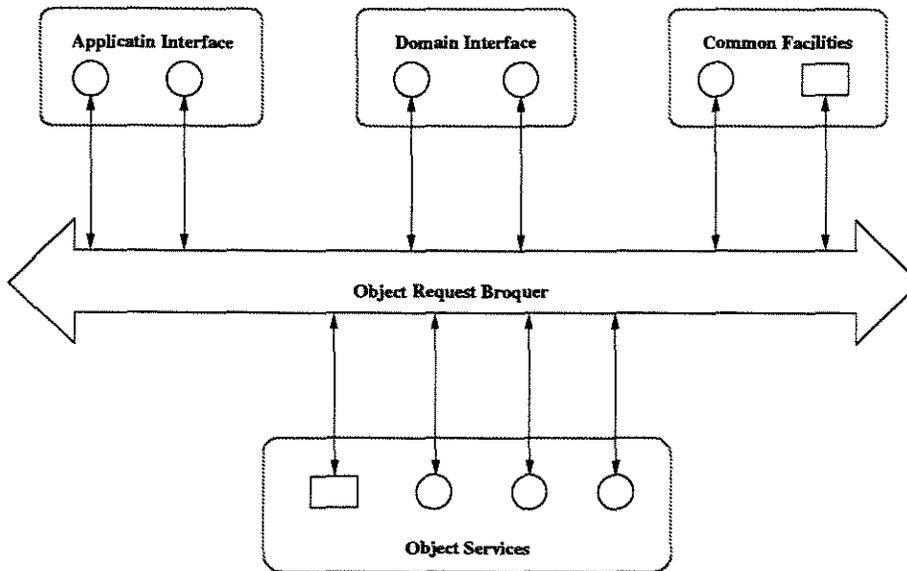


Figura 2.6: *Object Management Architecture*

As aplicações que desejam utilizar serviços de aplicações escritas no paradigma de Orientação a Objetos e que foram desenvolvidas sob a OMA necessitam apenas suportar as interfaces do OMG. Pelo Serviço de Objetos estas aplicações poderiam ser embutidas em objetos (*object wrappers*) que participam da OMA.

ORB

O ORB (*object request broker*) é o mecanismo que possibilita a troca de mensagens entre um cliente e um servidor, pela rede, de maneira transparente para ambos. Não é necessário que um cliente tenha informações específicas sobre o servidor como os mecanismos usados para se comunicar, ativar e localizar um objeto. Portanto o ORB é a base para a interoperabilidade entre objetos distribuídos e aplicações em ambientes homogêneos ou heterogêneos [SV98, AOS⁺00, OMG98a].

⁹componente responsável pelos serviços de negociação dos pedidos em plataformas distribuídas e heterogêneas, satisfazendo os requisitos de transparência de localização, transparência das plataformas *hardware e software* e garantida de entrega.

Serviços de Objetos

Serviços de Objetos são serviços de propósito geral utilizados para construir facilidades de alto nível e esquemas (*framework*) de objetos que possam interoperar através de ambientes multi-plataformas. O conjunto de Serviços de Objetos adotado pelo OMG, CORBAservices [OMG95, OOS01], inclui os serviços de: Nome, Eventos, Ciclo de Vida, Persistência de Objetos, Transações, Controle de Concorrência, Relacionamento, Licença, Consultas, Segurança, Tempo, Coleções e Negociação.

Facilidades Comuns

Facilidades Comuns são interfaces de propósito geral, aplicadas à maioria dos domínios das aplicações. Assim como os Serviços de Objetos, as Facilidades Comuns são definidas em termos de interfaces e suas semânticas as quais os objetos devem suportar para prover ou utilizá-las.

Interfaces de Domínio

São as interfaces de domínio próprio das aplicações como por exemplo: finanças, manufatura e telecomunicações.

2.5 Resumo do Capítulo

Este capítulo abordou quatro grandes temas: heterogeneidade dos bancos de dados, sistemas de informações legados, mapeamento de processos de negócio e a arquitetura da CORBA. Todos estes temas estão ligados à migração e integração de sistemas legados.

Em relação a bancos de dados heterogêneos foram detalhados os aspectos de heterogeneidade, autonomia e distribuição e as abordagens para integração de SDBH. As abordagens discutidas foram o mapeamentos de esquemas locais em esquemas componentes, federados e externo sob um modelo de dados comum, o processamento de consulta global e as diferentes formas de construção de SBDH utilizando-se mediadores. Foi apresentada a arquitetura de referência proposta por Sheth e Larson [SL90] e utilizada pela maioria dos trabalhos na área de SDBH.

Sobre o tema de Sistemas de Informação Legados foram analisados trabalhos que sugerem a migração incremental como a melhor maneira para migração destes sistemas para novos sistemas de informação. Alguns trabalhos lidam com os aspectos técnicos da migração como o detalhamento dos sistemas legados, os problemas de conversão de dados, a construção de sistemas de banco de dados federados para a integração dos sistemas.

Enquanto que outros trabalhos estão mais interessados nos aspectos de requisitos e necessidades das empresas e a forma de recolher estas informações para garantir uma correta especificação dos novos sistemas.

Outros temas explorados foram as metodologias e processos de mapeamento dos processos de negócio e reengenharia de processos. Por último foi analisada a arquitetura CORBA. Os componentes do modelo de objetos: objetos, atributos, operações, tipos e interfaces; e os componentes de arquitetura: ORB, Serviços de Objetos, Facilidade comuns e as Interfaces de Domínio.

Capítulo 3

Metodologia de Migração de Sistemas Legados

Este capítulo detalha uma metodologia de migração de sistemas legados para sistemas que possam atender as necessidades de negócio da organização. Estes sistemas devem ter pelo menos duas características: facilidade de adaptação às novas regras de negócio e facilidade de integração com outros sistemas. Para que esses sistemas possam atender as necessidades da organização, deve-se conhecer os requisitos de negócio através de um mapeamento dos processos de negócio da organização. Também serão detalhadas as possíveis arquiteturas de sistemas legados e será apresentado as características de uma arquitetura de sistemas de informação que suporte a integração e migração de sistemas legados.

3.1 Metodologia Proposta

Conforme visto no capítulo anterior, existem inúmeros trabalhos na área de migração de sistemas de informação legados. Nesta dissertação será especificada uma metodologia genérica que possa atender a migração dos sistemas legados e integração entre os mais diferentes sistemas de informação com os sistemas legados. Desta forma não serão indicados métodos ou ferramentas específicas de integração de SBDH ou alguma ferramenta ou arquitetura de interoperação de sistemas pois essas decisões são particulares para cada situação.

Na migração de grandes sistemas legados temos muitas dificuldades, como por exemplo: tradução, correção e integração dos dados. Para muitas atividades não existe nenhuma ferramenta específica e devem ser realizadas por técnicos que conheçam a tecnologia e tenham domínio da aplicação. A possibilidade de falhas é grande e a utilização de uma metodologia para guiar os passos da migração é muito útil para que o foco da migração não seja desviado ou mesmo alterado durante o processo de migração.

A metodologia *Chicken Little* será utilizada como base para a elaboração da metodologia proposta neste trabalho. Outros trabalhos importantes que também serão utilizados são os apresentados por Ganti et al. [GB95], Bergey et al. [BNS97] e Soares [Soa98].

A seguir são descritos aspectos relevantes para a metodologia a ser proposta: definição dos requisitos da migração, dos fluxos de informação da empresa, dos sistemas e ferramentas que serão utilizadas.

A maioria dos trabalhos existentes sobre migração de sistemas legados, com exceção de Ganti et al. [GB95] e Bergey et al. [BNS97], estão focados apenas no processo de construção dos novos sistemas [BS95], na integração dos dados [Soa98], na integração das aplicações [Win95, NEK94, MNB⁺94, Cha98] ou na conversão dos dados [Rob97].

As propostas de Ganti et al. [GB95] e Bergey et al. [BNS97] tratam do problema da organização como um todo, onde os novos sistemas legados serão incorporados dentro de uma estratégia global de sistemas de informação da organização. Este estudo não deve ser muito demorado nem detalhado mas servirá de base para a escolha dos sistemas ou subsistemas legados a serem migrados de acordo com as necessidades de negócio da organização.

Uma metodologia para fazer o levantamento dos processos de negócio da organização é apresentada por Jacobson et al. [JEJ95]. Eles utilizam o modelo de casos de uso e modelagem de objetos para fazer a análise e documentação dos processos da organização.

Outra atividade importante para o sucesso da migração é a definição de uma arquitetura de sistemas de informação da organização. Sem ela os novos sistemas migrados têm uma grande possibilidade de se tornarem os sistemas legados do futuro [NM95] [Gar95]. A simples migração dos sistemas legados pode não trazer os benefícios de integração total dos sistemas da organização, tão esperada por todos. Corre-se o risco de criar inúmeros sistemas desenvolvidos em uma linguagem de programação moderna, que utilizam SGBD comerciais, que podem ser executados via *internet*, mas que não conseguem acessar os dados de outro sistema para um relatório gerencial ou então o número de ferramentas pode crescer vertiginosamente se não houver uma padronização anterior às migrações.

É impossível conhecer todos os requisitos necessários para o novo sistema de informação que está sendo construído. O sistema deve ser projetado o mais flexível possível, para que novos requisitos possam ser incorporados, sem muitas alterações nos componentes já migrados [RHH98]. Por exemplo: o sistema não deve ser especificado para depender apenas de um sistema operacional, sistema gerenciador de banco de dados ou linguagem de programação; ele deve ser portátil para evitar a dependência de apenas um fornecedor, mesmo que isto traga um aumento no tempo de desenvolvimento ou mesmo diminuição do desempenho de algumas funções específicas do sistema destino.

Após o mapeamento dos processos da organização, feita a análise global dos requisitos, priorizados os sistemas legados a serem migrados, projetada a arquitetura de sistemas de

informação da organização, deve-se iniciar a migração dos sistemas legados de forma incremental. Nesta fase ocorrerá o detalhamento de cada sistema legado. Sempre que possível deve ser feito uma redução do volume de dados e funções dos sistemas legados de acordo com os requisitos globais da empresa. Desta forma, os riscos da migração são reduzidos pois segundo Brodie et al. [BS95] eles são diretamente proporcionais ao volume de dados e funções a serem migrados.

O último aspecto da migração dos sistemas legados é a implantação do novo sistema. Normalmente esta fase pode ser vista como mais uma das fases de migração de um componente do sistema legado mas dependendo do tipo do sistema, como por exemplo: sistemas bancários ou de telecomunicações, esta mudança pode envolver centenas ou milhares de usuários e máquinas em vários locais e pode durar algumas semanas e até meses. Esta fase pode ser considerada uma fase independente da migração do sistema legado, sendo assim ela também deverá ser feita incrementalmente, independente do trabalho de projeto e implementação dos componentes do novo sistema.

A metodologia proposta nesta dissertação possui três grandes fases:

1. Análise global - nesta fase será feito um estudo geral da organização, definição dos macro-processos dos quais os sistemas legados fazem parte, os problemas apresentados por estes sistemas e as novas necessidades do negócio. Esta fase pode ser tão extensa e detalhada quanto se queira, pois todas as grandes empresas têm muitos problemas na área de mapeamento de processos porém não deve-se perder o foco de que o objetivo principal é a migração dos sistemas legados;
2. Definição de uma arquitetura para os sistemas de informação - esta fase é extremamente técnica e deve se basear nos requisitos levantados durante a análise global para que seja possível a migração incremental dos sistemas legados, sem correr o risco de criar novos sistemas totalmente desintegrados. Caso a organização já possua uma arquitetura de sistemas de informação, ela deverá ser utilizada e se for necessário deve ser atualizada;
3. Migração incremental dos sistemas legados - esta é a fase mais complexa, onde os sistemas legados a serem migrados, selecionados durante a fase de análise global, deverão ser detalhados para posterior construção dos novos sistemas de acordo com a arquitetura de sistemas de informação definida na segunda fase. Nesta fase também serão abordados aspectos da implantação do novo sistema.

A figura 3.1 apresenta as fases da metodologia.

A metodologia apresenta as fases em sequência correta de execução porém a fase de migração dos sistemas legados é a que entra nos mínimos detalhes dos sistemas, sen-

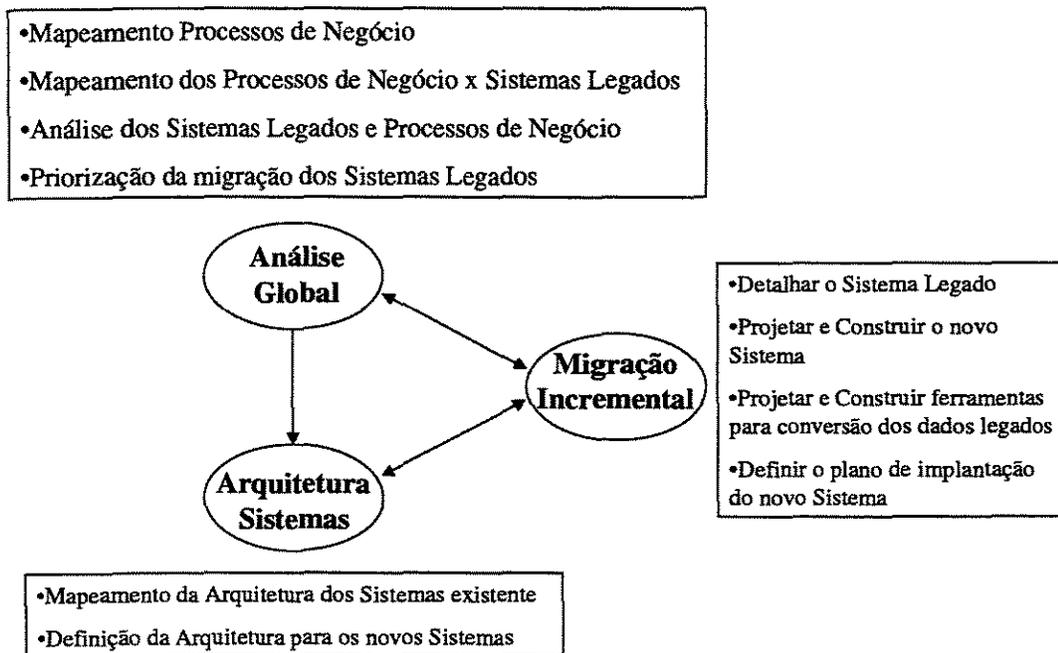


Figura 3.1: Metodologia para Migração de Sistemas Legados

do assim ela pode realimentar as duas fases anteriores com informações que não foram detectadas durante as análises iniciais. A seguir serão detalhada as fases da metodologia.

3.2 Análise Global

A fase de Análise Global dos processos de negócios e sistemas legados é a fase decisiva para o sucesso da migração dos sistemas legados. Nesta fase deve-se definir os processos que estão requisitando alterações nos sistemas legados, quais as alterações necessárias e a priorização das alterações. Esta fase da migração dos sistemas será dividida em quatro etapas:

1. Mapeamento dos processos do negócio;
2. Mapeamento dos sistemas de informação em relação aos processos do negócio;
3. Análise dos processos de negócio e sistemas legados;
4. Priorização dos processos e sistemas legados a serem migrados.

A seguir serão descritas cada uma das etapas.

3.2.1 Mapeamento dos Processos

Neste passo da metodologia será feita uma análise geral dos processos da organização. Algumas das razões para se entender a situação atual dos processos de negócio são:

- propiciar um entendimento comum dos processos e sistemas de informação que não estão funcionando bem. Isto se faz necessário para identificar as áreas que precisam ser alteradas;
- entender como se deve proceder para alterar os processos e sistemas de informação atuais de forma a satisfazer os novos objetivos e requisitos do negócio;
- dimensionar as alterações que vão ocorrer decorrentes de uma mudança no fluxo dos negócios.

Para mapear os processos de negócio será utilizado o modelo de engenharia reversa descrito por Jacobson et al. [JEJ95]. Eles utilizam os modelos de caso de uso e de objetos para fazer o mapeamento dos processos atuais da organização e propor a reengenharia dos processos. Neste trabalho estes modelos serão utilizados com o propósito específico de mapeamento dos processos e posterior mapeamento dos sistemas de informação utilizados

pela organização. Este método foi escolhido por utilizar o modelo de casos de uso e modelo de objetos para mapeamento de processos de negócio. Estes modelos também poderão ser utilizados na fase de especificação dos novos sistemas de informação por serem a base da metodologia UML (*Unified Method Language*) [FS97].

Nesta dissertação não serão definidos os modelos de caso de uso e modelo de objetos por estar fora do escopo deste trabalho. Eles serão apenas citados e utilizados de acordo com a metodologia descrita por Jacobson et al. [JEJ95]. Para maiores detalhes deve ser feito um estudo da metodologia UML [FS97].

Segundo Jacobson et al. [JEJ95] uma linguagem de modelagem de negócios deve permitir que se descreva os vários tipos de tarefas ou processos internos aos processos de negócio assim como a maneira pela qual esses processos internos interagem para oferecer um produto ou serviço para o cliente. O modelo para descrever os processos do negócio é o modelo de caso de uso ou modelo externo e o modelo utilizado para descrever os processos internos é o modelo de objetos ou modelo interno.

Para Jacobson et al. [JEJ95] tanto o modelo interno quanto o externo podem ser de dois tipos: ideal e real. O modelo ideal da companhia não leva em consideração que ela já possua uma estrutura funcional ou possa estar espalhada geograficamente entre várias filiais enquanto que o modelo real considera todos estes fatores. Nesta dissertação apenas o modelo real será utilizado pois o escopo deste trabalho é o mapeamento dos processos atuais para posterior migração dos sistemas legados da organização.

Modelo de Caso de Uso

Inicialmente deve-se fazer um modelo de caso de uso geral de toda a organização para que toda a equipe esteja de acordo em como o negócio está operando e para servir de ferramenta de negociação com a gerência da empresa. Após fazer este modelo geral, deve-se iniciar o detalhamento dos casos de uso.

A construção do modelo de casos de uso deve começar através da identificação dos atores que tipicamente são: os parceiros, fornecedores, órgãos governamentais e as subsidiárias. Caso alguma parte da empresa não esteja sendo mapeada nos casos de uso, ela deve aparecer como um ator no modelo.

Um critério para que alguma entidade seja um ator é que ela não deve pertencer ao negócio que está sendo modelado, ela apenas interage com o negócio.

Uma maneira de certificar-se que todos os atores foram identificados é estudar exemplos de pessoas ou grupo de pessoas com quem a companhia tem contato. Agrupe essas pessoas de acordo com o papel que ela realiza na companhia. Cada grupo corresponderá a um ator que deve ser incluído no modelo.

Após a identificação dos atores, deve-se iniciar a modelagem dos casos de uso associados a cada ator.

Segundo Jacobson et al. [JEJ95] um caso de uso é uma sequência de transações em um sistema cuja tarefa é permitir um resultado de valor mensurável para um ator individual do sistema. O valor mensurável está associado à idéia de que um caso de uso deve auxiliar o ator a executar uma tarefa que tem um valor que pode ser medido ou identificado. No caso específico de mapeamento de processos da companhia, os casos de uso são os processos de negócio.

Jacobson et al. [JEJ95] recomendam que os casos de uso devem ser pensados como se fossem máquinas de eventos de estado no qual o estado de uma instância de um caso de uso representa qual estímulo foi recebido para ter alcançado o estado atual. Da mesma forma, um estímulo pode fazer com que o caso de uso deixe seu estado atual, depois de ter completado a transação, para adotar um novo estado.

Caso sejam identificados casos de uso que não estão conectados a nenhum ator, deve-se fazer uma verificação mais detalhada para certificar-se da real existência destes casos de uso. Nesta fase não se deve perder muito tempo tentando encontrar uma visão perfeita dos negócios. Deve-se prosseguir quando os fluxos mais importantes do negócio foram mapeados.

Para verificar se todas as responsabilidades do negócio estão incluídas no modelo de caso de uso, deve-se certificar que para cada responsabilidade existe pelo menos um caso de uso no modelo.

A descrição dos casos de uso deve ser feita utilizando-se uma linguagem natural. Deve-se descrever em detalhes os fluxos de eventos do caso de uso e como eles interagem com os atores. Os fluxos alternativos e de exceção devem ser detalhados no mínimo para os casos de uso mais importantes. Os casos de uso devem representar um fluxo completo de eventos, isto é, contém início e fim de estados bem definidos.

A descrição de um caso de uso deve conter as seguintes informações:

- uma lista de atores associados a ele;
- uma lista de casos de uso associados a ele;
- uma descrição detalhada dos fluxos básico e alternativos do caso de uso;
- uma descrição detalhada dos fluxos opcionais e de exceção.

Por fim deve ser feita a revisão do modelo (atores, casos de uso e suas associações) e dos casos de uso individualmente. Todas as pessoas envolvidas na elaboração dos casos de uso e as pessoas responsáveis pela desenvolvimento das ferramentas de suporte computacional devem participar da revisão.

Antes de iniciar a revisão formal do modelo deve-se verificar os seguintes pontos:

- se a comunicação para e/ou do ator está correta;

- se os fluxos de evento estão descritos em um nível de detalhe suficiente;
- se os fluxos de evento estão corretos e completos;
- se é possível simplificar e agilizar o fluxo dos eventos;
- se todos os fluxos alternativos estão descritos.

A construção de um modelo de casos de uso que seja efetivo é difícil. Segundo Jacobson et al. [JEJ95] a maioria dos negócios podem ser descritos com 15 ou 20 casos de uso principais. Davenport [Dav93] também considera que cerca de 20 processos são suficientes para descrever o negócio. Normalmente deve-se desenvolver modelos de casos de uso para as áreas chave dentro da empresa. Estas áreas são as que têm contato direto com os clientes e aquelas que fazem com que a empresa produza.

Modelo de Objetos

Após os casos de uso terem sido descritos, deve-se fazer um modelo de objetos do negócio. O modelo de objetos é necessário pois o modelo de caso de uso não fornece uma figura de como o negócio está estruturado internamente para atender seus requisitos, não mostra como os processos de negócio são executados através de várias atividades e como estas atividades estão correlacionadas. Também não consegue mostrar os produtos que são usados ou produzidos pelo negócio ou quais recursos devem ser utilizados para implementar as várias atividades.

O modelo de objetos irá descrever como o novo negócio irá oferecer os casos de uso para o ambiente, isto é, descreve a operação do negócio. Ele deve considerar as restrições do negócio, como:

- distribuição geográfica do negócio - centralizada ou espalhada;
- os empregados nem sempre tem a competência para realizar as tarefas requeridas;
- conviver com o suporte computacional antigo por algum tempo até que seja totalmente renovado;
- razões não técnicas, isto é, políticas.

Para aumentar o entendimento do modelo de objetos deve-se utilizar três tipos de objetos: interface, controle e entidade. Os objetos de interface e controle geralmente representam tarefas do negócio enquanto que os objetos de entidade representam produtos, documentos e ações diárias ou requeridas por órgãos de regulação.

Para cada interação entre os atores e os casos de uso pode-se identificar um objeto de interface. Os objetos de controle são identificados no fluxo internos dos eventos. Os objetos de entidade podem ser encontrados ao analisar cada fluxo de evento dos casos de uso.

Ao descrever cada objeto deve-se verificar qual o papel do objeto em cada caso de uso em que ele participa. Cada papel é descrito na forma de operações e/ou atributos. O essencial é que cada papel de cada caso de uso seja executado por um objeto.

Na modelagem de objetos são identificados os subsistemas, isto é, grupo de objetos, e os casos de uso que fazem a comunicação dos subsistemas. Existem casos em que deve-se detalhar um pouco mais, chegando no nível de casos de uso entre objetos dentro dos subsistemas ou entre subsistemas diferentes. Tudo isto deve ser muito bem avaliado para que não acabe onerando muito a modelagem com tantos detalhes.

Diagramas de Interação

O diagrama de interação identifica a interação entre os fluxo de eventos de cada caso de uso com os objetos que participam do caso de uso. Desta forma os papéis e responsabilidades dos objetos ficam claras. Cada fluxo alternativo do caso de uso deve ser mapeado em um diagrama de interação. Neste diagrama devem ser identificadas as operações que representam as responsabilidades do objeto e os seus parâmetros.

3.2.2 Mapeamento dos Sistemas de Informações

Neste passo devem ser identificados os sistemas de informação existentes na organização. Os sistemas de informação devem ser analisados em relação às suas plataformas de *hardware* e *software*. Deve-se descrever como a utilização do sistema otimiza o trabalho e deve-se discutir as vantagens e desvantagens do seu uso. Neste passo pode-se perceber as limitações dos sistemas atuais da empresa e a necessidade de mudanças ou migração para novos sistemas.

O mapeamento entre os processos do negócio e os sistemas de informação existentes na companhia deve ser feito utilizando-se uma matriz de mapeamento (MM) que deve ser formada colocando-se os Sistemas de Informação nas linhas da matriz e os processos do negócio nas colunas. Neste mapeamento deve-se assinalar a correspondência existente entre os processos do negócio e os sistemas de informação. A tabela 3.1 apresenta a matriz de mapeamento.

	Processo P1	Processo P2	...	Processo Pn
Sistema S1	X	X		
Sistema S2	X			X
...				
Sistema Sn	X	X		X

Tabela 3.1: Matriz de Mapeamento: Processos X Sistemas

3.2.3 Análise dos Processos e Sistemas Legados

Após conseguir mapear o modelo atual do negócio, deve-se analisar as especificações para priorizá-los. A profundidade da análise deve ser definida para cada situação. Os possíveis passos a serem seguidos são:

- percorrer cada caso de uso classificando-o como: com valor adicionado, sem valor adicionado;
- identificar os problemas e limitações dos sistemas legados em cada caso de uso;
- investigar a possibilidade de melhoria, a curto-prazo, dos sistemas legados nos casos de uso existentes.

3.2.4 Priorização dos Processos e Sistemas Legados

A seguir deve-se priorizar os processos que serão reprojutados. A equipe de reengenharia dos processos e migração dos sistemas legados deve começar o trabalho logo após a priorização.

A priorização deve focar as áreas que irão prover resultados positivos rapidamente. Hammer et al. [HC93] descrevem como selecionar os processos mais adequados para serem reprojutados.

- identifique aqueles que têm mais problemas;
- identifique aqueles que têm maior impacto nos clientes;
- identifique os que têm maior potencial para melhorias radicais dos negócios;
- identifique os que podem ser melhorados fácil e rapidamente;
- identifique todos os que podem obter as maiores melhorias internas nos negócios.

Após a priorização deve-se descrever mais detalhadamente os casos de uso que foram priorizados, deixando os menos prioritários para serem detalhados a médio ou longo prazo.

3.3 Arquitetura de Sistemas de Informação

Após ter sido feito o mapeamento dos processos da organização e dos sistemas de informação atuais deve-se fazer um levantamento da arquitetura atual dos sistemas legados e, de acordo com as necessidades, projetar uma arquitetura para os novos sistemas de forma a poder atender os requisitos da organização.

Segundo Garlan [Gar95] os grandes problemas no desenvolvimento de grandes sistemas de informação são a organização, em um alto nível, dos elementos computacionais e a interação entre eles. Exemplos destes elementos são: protocolos de comunicação, sincronização e acesso de dados; distribuição física dos sistemas e dados, modelo de dados, linguagens de programação, SGBD, métodos de modelagem do sistema, mediadores de comunicação entre as aplicações e os SGBD, escalabilidade e desempenho. Desta forma, os novos sistemas que serão desenvolvidos para substituir os sistemas legados devem fazer parte de uma estrutura global de toda a empresa.

Uma consideração chave no processo de migração dos sistemas legados é a capacidade de continuar co-processando os resíduos dos sistemas legados enquanto alguns sistemas já foram parcialmente migrados. Este co-processamento será possível através da utilização de mediadores.

As funções desempenhadas por um mediador são:

1. encapsulador de certos módulos das mudanças que estão acontecendo em outros. Esta funcionalidade vai permitir que as mudanças possam ocorrer incrementalmente
2. tradutor de pedidos e dados entre os componentes que ele está mediando. Por exemplo, o mediador pode traduzir chamadas da interface de usuário para o sistema destino ou dados do sistema destino para o formato da interface de usuário do sistema legado
3. coordenador entre os componentes que estão sendo mediados, isto é, coordenar as ações de consulta e atualização, decompondo-as e enviando as subpartes para os componentes do sistema destino ou do sistema legado, que estão tratando aquele dado no momento. A mediação de uma atualização é mais complexa uma vez que, ao final da operação de atualização, todas as cópias dos dados devem refletir a atualização, mantendo os bancos de dados destino e legado consistentes.

A definição de qual tipo de mediador deve ser utilizado depende da arquitetura do sistema legado e da arquitetura do sistema final. Para cada situação existe um tipo de mediador a ser utilizado.

As ferramentas comerciais disponíveis para fazer a mediação entre sistemas não são do tipo *plug&play*. Algumas se propõem a fazer mediação entre alguns bancos de dados mas

existe um trabalho muito grande de configuração principalmente em relação à semântica dos dados. Recentemente começaram a surgir ferramentas comerciais que dão o suporte para que se possa construir os mediadores de aplicações e de sistemas de informação de forma transparente em relação ao *hardware*, sistema operacional e linguagens de programação utilizados pelos sistemas de informação. Porém estas ferramentas apenas dão o suporte para que os mediadores sejam construídos.

Esta fase da migração dos sistemas legados é composta de duas etapas:

- Mapeamento da arquitetura dos sistemas de informação existentes;
- Definição dos componentes de *software* que compõem uma arquitetura de sistemas de informação.

A seguir serão detalhadas as duas etapas.

3.3.1 Arquitetura de Sistemas Legados

Além da metodologia a ser utilizada na migração do sistema legado, é necessário conhecer a arquitetura do sistema que está sendo migrado para que se possa dimensionar o esforço necessário em todo o processo, qual método aplicar e a ordem em que os módulos serão migrados. Através da arquitetura dos sistemas legados é possível identificar a época em que eles foram projetados e a forma como eles foram evoluindo ao longo do tempo. Os sistemas foram se tornando mais complexos à medida que novas técnicas eram aplicadas na sua construção e novos requisitos eram necessários. A decomposição funcional, linguagem de programação estruturada e mais recentemente sistema gerenciadores de banco de dados, são ferramentas que apenas começaram a ser utilizadas em larga escala no desenvolvimento de sistemas de informação a partir da década de 1980. A medida que estas novas técnicas e ferramentas foram introduzidas nos sistemas já em produção sem que eles passassem por uma completa reestruturação, fez com que eles adquirissem as características de um sistema com uma arquitetura híbrida descrita a seguir.

Os sistemas de informação podem ser divididos, em relação à sua organização interna, em três grandes componentes: interfaces, aplicações e banco de dados. Segundo Brodie et al. [BS95] a arquitetura dos sistemas de informações pode ser classificada, conforme a organização dos seus componentes internos, em:

1. não estruturada: nenhum componente funcional pode ser separado. O sistema todo é visto como uma caixa preta (fig. 3.2a);
2. semi-estruturada: apenas as interfaces do sistema e do usuário estão separadas do restante do sistema (fig. 3.2b);

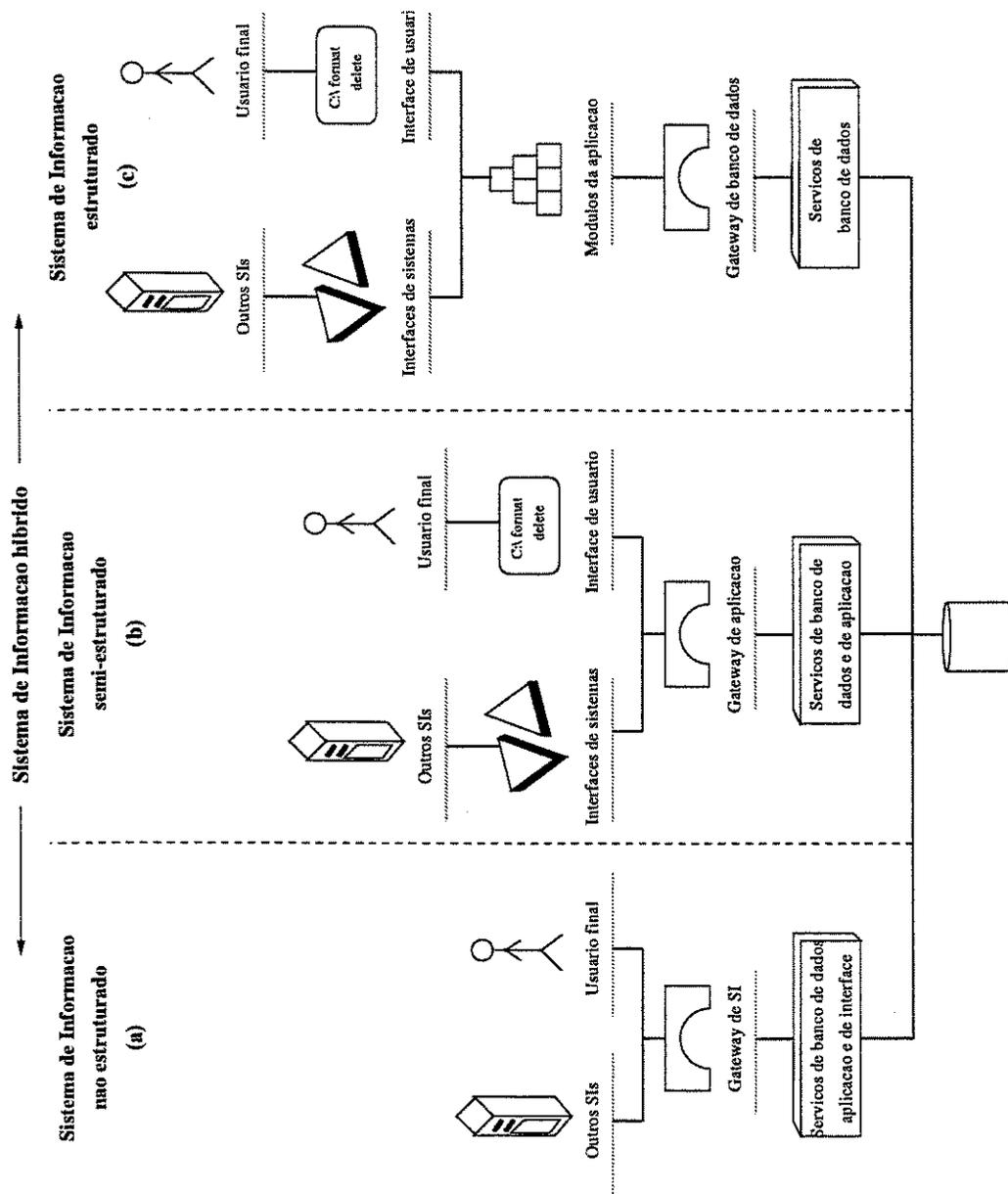


Figura 3.2: Arquitetura de Sistemas de Informação

3. estruturada: as interfaces, aplicações e serviços de banco de dados estão em componentes distintos com interfaces bem definidas. Esta é a melhor arquitetura para migrar um sistema de informação (fig. 3.2c);
4. híbrida: existem partes dos sistema que podem ser decompostas e partes que atuam como caixas-preta. Este tipo de sistema decorre de anos de evolução, sempre reaproveitando o código existente (fig. 3.2).

Quanto menos estruturado o sistema de informação legado, mais difícil e suscetível a erros se torna a migração.

Durante o processo de migração incremental dos sistemas legados, proposto pelo método *Chicken Little*, o sistema final e o sistema legado formarão um sistema composto¹ durante toda a fase de migração. Nesta fase os dois sistemas estarão conectados via mediadores, os quais desempenham um papel importantíssimo durante toda a fase de migração (fig. 3.2).

Como já visto, o mediador desempenha um papel muito importante na migração de um sistema e o local em que ele estará, também pode causar um impacto muito forte na arquitetura de migração. De acordo com o seu posicionamento na arquitetura do sistema composto, os mediadores se classificam em: mediador de banco de dados, mediador de aplicação e mediador de sistema de informação. Esta classificação está diretamente relacionada à arquitetura do sistema legado, isto é, sistemas estruturados podem ter um mediador de banco de dados, sistemas semi-estruturados podem ter mediador de aplicação e sistemas não estruturados podem ter mediador de sistema de informação (fig. 3.2).

Brodie et al. [BS95] ainda dividem os mediadores em relação ao fluxo das informações, em duas partes: *forward gateway* que permite ao sistema legado acessar dados na parte já migrada do sistema e *reverse gateway* que permite ao sistema destino acessar dados no gerenciador de dados do sistema legado (fig. 3.3).

O tempo em que um mediador permanece no sistema, está relacionado com o tempo necessário para migração do componente do sistema legado, o qual ele faz a intermediação para o sistema destino. Na prática os mediadores podem se tornar parte permanente da arquitetura do sistema composto pois muitas partes podem não ser migradas, por fatores econômicos ou até tecnológicos, por exemplo.

3.3.2 Características de Arquitetura de Sistemas de Informação

No passo anterior foi feito um levantamento e classificação dos sistemas legados. Este trabalho já faz parte da especificação da arquitetura global da organização porque estes

¹sistema composto é o sistema que apresenta alguns componentes migrados para a nova arquitetura e outros ainda no formato do sistema legado

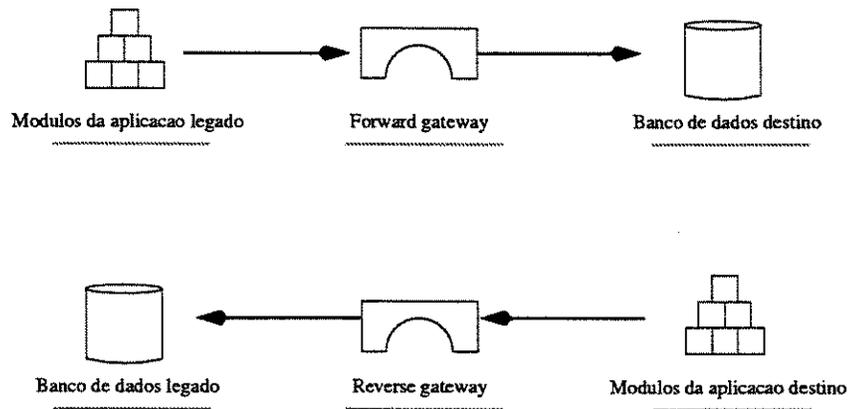


Figura 3.3: *Forward and Reverse gateways*

sistemas não irão desaparecer da noite para o dia. De acordo com a metodologia *Chicken Little* os sistemas legados irão permanecer durante todo o processo de migração.

Neste passo será especificada a arquitetura dos novos sistemas de informação da organização, caso não exista uma. Nesta arquitetura além de serem especificados os componentes dos novos sistemas, deverão ser especificados componentes que farão a integração dos novos sistemas com os sistemas legados e vice-versa.

Uma arquitetura é o conjunto de todos os componentes tecnológicos que juntos suportam a organização. Essas tecnologias nunca foram coordenadas, suas compras nunca foram planejadas simultaneamente e os componentes não trabalham harmoniosamente. A arquitetura que prevalece atualmente é a fracamente acoplada de vários *hardwares*, *softwares* e redes.

Não se pode prever as necessidade futuras dos requisitos dos negócios ou os avanços da tecnologia mas uma boa escolha da arquitetura global do sistema pode facilitar futuras migrações. A migração dos sistemas legados de forma incremental, pode ser atacada em cada um dos níveis da arquitetura do sistema destino por equipes diferentes, cada uma com as suas habilidades e competências.

Segundo Garlan [Gar95], a utilização de uma arquitetura de *software* pode ter um impacto positivo em pelo menos cinco aspectos do desenvolvimento de *software*.

1. *Compreensão* - Arquitetura de *software* simplifica a compreensão de grandes sistemas de informação pois o sistema pode ser visto em um alto nível de abstração, isto é, apenas seu grandes componentes. Além de expor as restrições de alto nível do sistema, tornando mais fácil a decisão de escolher ou trocar alguma peça do sistema;
2. *Reuso* - O uso de descrição de arquiteturas possibilita não apenas o reuso de funções

ou classes em bibliotecas de uso geral mas também possibilita o reuso de grandes componentes de *software*, padrões, *frameworks*;

3. *Evolução* - A arquitetura de *software* pode expor as futuras evoluções do sistema. Colocando as interfaces bem definidas entre os componentes os programadores podem entender melhor as ramificações das mudanças e auxiliar na estimativa correta do custo. Pode-se portanto trocar um componente do sistema, por motivos de desempenho ou interoperabilidade, sem afetar os demais, desde que as interfaces continuem as mesmas;
4. *Análise* - A descrição da arquitetura cria uma nova oportunidade de análise, incluindo formas de verificação de consistência do sistema em um alto nível, conformidade com os atributos de qualidade e conformidade com os padrões de arquitetura global da organização;
5. *Gerenciamento* - A possibilidade de especificar os requisitos de capacidade de operação inicial do sistema bem como a dimensão de crescimento antecipadamente faz com que o risco do desenvolvimento do sistema não atender os requisitos seja pequeno, além de poder acomodar mudanças com maior facilidade.

Para Ganti et al. [GB95] uma arquitetura de sistemas deve ser capaz de produzir sistemas de informação que sejam:

- funcionais - o sistema deve fazer o que está proposto, deve atender os objetivos para os quais foi especificado. Deve também atender as necessidades de desempenho, segurança, disponibilidade e consistência;
- adaptáveis - os sistemas devem permitir constantes melhorias, seja por mudanças na tecnologia - novos fornecedores, produtos e ferramentas, ou dos usuários ou ainda dos processos do negócio;
- atrativos - o sistema deve ser agradável e estimular o usuário a utilizá-lo. Os erros e situações anormais devem ser contornadas de maneira amigável e sempre que possível através de mensagens de fácil entendimento.

A arquitetura deve dar as diretrizes para a construção de sistemas que satisfaçam os seus usuários e para que os objetivos da empresa sejam atingidos.

Os sistemas de informação que atingem seus objetivos apresentam três conceitos essenciais:

1. **Abstração** - situações específicas são abstraídas para situações gerais. Isto faz com que o sistema pareça simples e intuitivo para o usuário e funções mais complexas podem ficar transparentes. Para os desenvolvedores abstração significa independência entre a especificação e a implementação do sistema;
2. **Integração** - as diversas partes do sistema devem ser capazes de trocar informações e conhecimento, o que é uma oportunidade para cooperação;
3. **Extensibilidade** - os sistemas não podem ser uma fotografia no tempo, eles devem ser continuamente estendidos.

Entre todos os pontos da arquitetura para a transição para sistemas distribuídos, o balanço entre autonomia local e integração global provavelmente é a mais difícil de ser decidida. Os requisitos do negócio devem estabelecer as regras para este balanço.

O primeiro passo para se definir a arquitetura do sistema de informação é definir o objetivo que a organização quer alcançar em relação ao sistema.

Também neste primeiro passo deve-se definir os requisitos gerais dos sistemas de informação da organização, como:

- segurança - mecanismos de autenticação, autorização e confidencialidade de dados;
- desempenho - paralelismo, *bandwidth*, caching;
- recuperação de falhas - automática ou backup;
- disponibilidade - gerenciamento de replicação;
- conectividade - tipos de mediadores que serão utilizados;
- integridade dos dados;
- escalabilidade - baseado na capacidade de processamento do *hardware* ou distribuição do processamento;
- transparência - modelo cliente/servidor, arquitetura *3-tier* e serviços distribuídos.

O segundo passo é a definição dos componentes tecnológicos que irão formar a arquitetura de sistemas de informação da organização. Estas definições serão baseadas nos objetivos e restrições detalhados no passo anterior, além da arquitetura atual dos sistemas legados.

A seguir serão detalhados os principais componentes que fazem parte de uma arquitetura computacional de uma corporação [GFKH99].

Plataforma de *hardware*

A plataforma de *hardware* a ser utilizada deve ser definida no início da definição da nova arquitetura dos sistemas de informação pois este é um requisito limitante para muitos sistemas comerciais. Os fatores mais importantes que devem ser considerados para a escolha são:

- necessidade de processamento;
- necessidade de alta-disponibilidade;
- levantamento dos sistemas atuais, principalmente os que não serão migrados;
- orçamento para compra de máquinas e sistemas;
- custo para suporte e manutenção das máquinas;

Para efeito de facilidade de gerenciamento dos equipamentos o ideal é que haja uma padronização em termos de *hardware* porém muitas vezes esta padronização não é alcançada devido à grande heterogeneidade de sistemas que compõem a arquitetura da empresa.

Por outro lado para os sistemas que estão sendo desenvolvidos é muito importante que eles sejam projetados de modo a ficarem o mais independente possível da plataforma de *hardware*, não causando uma barreira no momento da sua aquisição e implantação.

Sistema Operacional

Normalmente cada fabricante de *hardware* no padrão Unix convencional possui a sua própria implementação, seja ela BSD ou *System V*. Atualmente a grande novidade que se encontra disponível comercialmente é o sistema operacional Linux para ser executado em computadores padrão IBM PC onde o sistema DOS/Windows é absoluto desde o início da fabricação destes equipamentos.

Já existe iniciativas de empresas fabricantes de *workstations* padrão Unix para rodar sob o sistema operacional Linux.

A escolha de sistemas que possam rodar nos diferentes tipos de sistemas operacionais é muito importante pois pode resultar em grande economia para as empresas. Da mesma forma novos sistemas devem ter este requisito na sua concepção.

Servidor Web

Existem servidores Web que são independentes de plataforma *hardware* e *software*, outros estão presos a apenas um fabricante, outros que executam associados ao banco de dados e outros que fazem parte da arquitetura de integração de aplicações da empresa. Cada

uma tem seus pontos fortes e fracos e eles devem ser analisados para que a escolha seja a melhor para as necessidades da arquitetura de sistemas de informação da companhia.

Topologia de Rede

A definição da topologia de rede a ser utilizada na solução global dos sistemas de informação da empresa depende de inúmeros fatores dentre eles:

- distribuição geográfica da companhia;
- localização dos servidores de dados;
- localização dos servidores de aplicação;
- localização dos usuários;
- volume de dados a ser trafegado;
- necessidade de desempenho.

Estes itens auxiliarão na definição do tipo de rede (LAN ou WAN), quantidade de banda, tipo de roteadores e *switches*.

A definição da topologia de rede parece na maioria dos casos estar longe da definição dos sistemas de informação. Pensar desta forma não é correto atualmente pois apesar do nível de rede estar abstraído do nível dos sistemas de informação, o seu desempenho está intimamente ligado a forma como o sistema atua (*desktop* ou *web*), a forma como ele acessa os dados no SGBD, o tipo de informações que ele manipula (dados convencionais ou dados não convencionais), distância dos usuários aos servidores de dados e aplicação versus a topologia de rede, quantidade de banda disponível e latência de rede.

Para a maioria das grandes empresas atuais é inviável economicamente construir suas próprias redes de dados. Normalmente elas têm comprado este tipo de serviço das empresas prestadoras de serviços de telecomunicações.

Gerenciador de Transação Distribuída

Se pensarmos no propósito dos bancos de dados podemos resumi-los a simples colecionadores e compartilhadores de dados ou estados. Quando se faz necessário alterá-los, existe um pequeno número de fenômenos básicos que são intuitivamente esperados para serem manipulados. Uma mudança completa deve persistir. Caso vários agentes façam mudanças simultaneamente, eles não devem inadvertidamente desfazer as mudanças de outros. Se múltiplas partes de um único fato são mudadas, como linhas de múltiplas tabelas, elas não devem inadvertidamente tornar-se inconsistentes devido a apenas parte

das mudanças serem feitas. Os serviços que garantem essas características de mudanças são chamados gerentes de transações [Spe89].

De um modo geral, uma transação é uma coleção de ações que modificam o estado de um ou mais objetos. Em sistemas de banco de dados convencionais, transações devem exibir três propriedades fundamentais - atomicidade, serialização e persistência² [GR93].

Serviços de gerencia de transação são invocados pelo gerente de transação para gerenciar uma *unidade de trabalho* ou *transação*. Uma transação está delimitada por um início e um final de trabalho. Se o invocador foi satisfeito, i.e., todo o serviço foi realizado, o invocador envia um pedido ao gerente de transação para tornar o trabalho permanente. Caso o trabalho não tenha sido completado pelo invocador, ele envia um pedido ao gerente de transação para abortar a transação. Com um gerente de transação distribuído, a complexidade de sites remotos a agentes independentes são transparentes para o invocador [GR93].

Para garantir essas características das transações, os gerentes de transações devem ser capazes de se recuperarem após falhas na transação, no servidor de processos, na máquina em que o processo está rodando e nos dispositivos de armazenagem. Quando os dados estão distribuídos entre múltiplos gerentes de recursos, novas fontes de falhas surgem. Um entre os vários gerentes de recursos pode falhar, a máquina hospedeira de um dos gerentes de recursos pode falhar independente das outras máquinas que gerenciam outros recursos, a comunicação entre as máquinas pode falhar [CKRZ98].

A gerência de transação é a habilidade de gerenciar o estado geral dos vários componentes. Gerenciadores de transação distribuída adiciona significativamente nova complexidade porque existem vários e diferentes tipos de componentes para gerenciar e tipos de falhas para serem recuperadas.

Em um ambiente de processamento distribuído, especialmente em ambientes heterogêneos, essas propriedades são extremamente difíceis de garantir. As ações podem ser executadas por processos executando em múltiplos sistemas que utilizam abordagens diferentes para garantir as propriedades das transações. Por exemplo, um sistema pode utilizar um protocolo de *locks* para garantir a serialização enquanto outro pode usar mecanismos de *time stamps*. O gerenciador de transação global deve ser capaz de unificar as diferenças de cada gerenciador para garantir a serialização global. Porém o pior caso é quando um dos sistemas que participa na transação distribuída não tem certa funcionalidade para assegurar as propriedades globais [GR93].

Assegurar a consistência³ em um ambiente de múltiplos bancos de dados sempre requer fortes restrições na autonomia dos bancos de dados locais, incluindo como e quando

²Estas propriedades também são conhecidas como ACID - atomicidade, consistência, isolamento e durabilidade

³Uma fonte de dados é dita consistente quando ela não contém contradições.

as operações são executadas e *como* e *quando* a comunicação com outros sistemas vai acontecer.

Em um sistema complexo é natural adotar uma estratégia na qual partes simples do sistema são modernizadas e distribuídas primeiramente, antes de lidar com partes mais complexas do sistema. Funções de consultas e relatórios são mais simples em relação a funções de atualização e, portanto, são geralmente mudadas em primeiro lugar. Sendo assim, a melhor maneira de proceder com a transição é fazer a transição dos componentes de consultas sem criar um impacto negativo nos componentes de transação. Então um dos fatores críticos de sucesso na transição de sistemas legados para uma arquitetura distribuída é a transição incremental para um processamento de transação distribuído [GB95].

Gerenciadores de Replicação

O gerenciamento da replicação pode ser feito com vários graus de transparência, dependendo do nível de abstração com que ele foi especificado. Idealmente deve ser especificado como uma unidade conceitual, i.e., uma visão conceitual do dado, definindo entidades, atributos e relacionamentos e pela definição de restrições conceituais que formam as regras do negócio. A maneira mais usada normalmente é uma abstração em um nível um pouco mais baixo, por exemplo, se os dados estão em um SGBD relacional, a replicação é especificada em termos de tabelas, chaves e integridade referencial, se o SGBD for orientado a objetos, a replicação é especificada em termos de tipos de dados. Em um nível mais baixo de abstração, a replicação pode ser especificada em termos de unidades físicas como registros, campos ou até bytes.

Quando as atualizações são feitas de um *site* remoto, existem severas restrições ao desempenho relativas ao gargalo da comunicação via rede. A cópia primária pode ter que mudar dinamicamente de local para se ajustar aos requisitos de desempenho, porém esta troca dinâmica é um desafio técnico muito grande.

Diante de tantas combinações de requisitos as tecnologias comerciais ainda estão imaturas e um longo caminho deve ser percorrido até fornecerem soluções mais adequadas. Alguns exemplos de gerenciadores de replicação comerciais são o NFS (*Network File System*) e o Sybase *Replication Server*.

SGBD

Banco de dados são o modo mais fácil de centralizar a definição e manutenção dos dados e estabelecer a independência entre as aplicações e os dados. A distribuição do ambiente introduz novos requisitos de centralização e transparência que vão além da capacidade dos bancos de dados tradicionais [EN94, DD99]. Pensando nisto os fabricantes de SGBD

introduziram novas características nos SGBD tradicionais, como:

- *storage procedures*: algumas operações básicas como restrições de integridade são armazenadas diretamente no banco de dados e disparada através de gatilhos;
- SGBD distribuídos: os gerenciadores padrões são acrescidos de funcionalidades que permitam acessar dados geograficamente dispersos. A função crítica neste caso é gerenciar as transações distribuídas e concorrentes. Atualmente os SGBD distribuídos são capazes apenas de gerenciar estas transações dentro do seu próprio ambiente, isto é, não são capazes de gerenciar transações distribuídas em SGBD heterogêneos;
- SGBD orientados a objetos ou objeto-relacional: algumas aplicações devido a sua complexidade ou mesmo fatores de desempenho se adequam melhor ao modelo orientado a objetos [DD00];
- tipos abstratos de dados - com o aumento do volume de informações armazenadas em bancos de dados e o tipo de informação que circula na Web é necessário que os gerenciadores possam armazenar dados não convencionais e não estruturados como imagens, texto, voz, etc;
- operadores espaciais - o aumento e maior disponibilidade de dados georeferenciados tem sido uma constante. A capacidade para armazenamento e gerenciamento de informações espacializadas pode ser um requisito para muitas aplicações;
- tipos de transação - os gerenciadores tradicionalmente são ótimos para controlar grandes quantidades de transações curtas porém para algumas aplicações é essencial a capacidade de gerenciar transações longas [DGMa95], [LG01].

Mediadores de Dados

Mediadores de dados são dispositivos para conectar sistemas heterogêneos. Na forma mais simples, são apenas tradutores de diferenças de implementação. Na forma mais complexa, podem traduzir as informações de um paradigma ou modelo de dados para outro. Eles podem ser vistos como componentes de *software* que encapsulam (*wrappers*) as diferenças de modelo de dados ou implementação. Os mediadores de dados executam os seguintes serviços:

- tradução de dialetos de SQL e representações de dados;
- padronização de protocolos de acesso de dados remotos;
- padronização de protocolos de comunicação Cliente/Servidor;

- mascarar diferenças dos SGBD;
- transparência de localização;
- controle de acesso e mecanismos de autenticação.

Eles normalmente têm as seguintes características:

- são limitados a bancos de dados relacionais;
- provêm um modelo de dados único para as aplicações;
- provêm abstração de dados e integração estáticas;
- provêm uma visão relacional a dados não relacionais;
- utilizam tradutores de SQL.

Se o cliente é uma ferramenta comercial, então a interface deve ser padronizada. Se o cliente é uma ferramenta especificamente desenvolvida, então o programador pode utilizar vários padrões disponíveis como ANSI SQL [MS92, DD96] para o conteúdo de mensagens, SAG, ODBC e ODAPI para troca de mensagens. Padrões de representação de dados são variados.

Serviços de Distribuição de Aplicação (*Middleware*)

Envio de mensagem é a base para a comunicação em computação distribuída. Inclui-se no envio de mensagens a tarefa de gerenciamento como: ativar o recipiente para pedir uma resposta [OMG98b].

Existe uma variedade de maneiras e inúmeros fatores necessários para realizar o envio de mensagens. Geralmente existem mecanismos para esconder a complexidade e as diferenças de comunicação independentemente se os módulos estão na mesma máquina ou em máquinas diferentes ou se os módulos estão no mesmo pacote de *software* ou não. Estes são fatores importantes pois após isso, outros fatores devem ser considerados, caso contrário, eles se confundirão com o mecanismo de troca de mensagem. Por exemplo: deve-se considerar mecanismos para interação entre aplicação e bancos de dados remotos. Se o mecanismo de mensagens estiver bem definido e encapsulado, pode-se examinar outros níveis de comunicação sem se importar com os baixos níveis de implementação.

Além da troca de mensagens outros pontos devem ser levados em consideração durante a comunicação entre sistemas. Deve existir um ambiente especializado para gerenciar o destino e conteúdo das mensagens.

Uma das formas mais comuns da arquitetura cliente/servidor é o acesso de dados remotos através de bancos de dados relacionais. A maioria dos mecanismos de troca de

mensagens são gerenciados transparentemente. A interação entre o usuário e o banco de dados envolve a troca de informação e respostas em uma forma padrão, isto é, existe um protocolo para recuperar linhas das tabelas de uma forma sistematizada.

O nível de acesso aos dados remotos provê os mecanismos de manipular, por exemplo, o tamanho e tipo dos dados que as consultas se referem.

Desta maneira as aplicações podem usar bancos de dados locais ou remotos, sem se preocupar com os mecanismos de distribuição.

Existem outros mecanismos de processamento distribuído como gerenciamento de transações distribuídas, *gateways* ou mediadores, bibliotecas de interface para gerenciamento de dados.

No ambiente de programação orientado a objetos, os objetos têm dois lados - interface e implementação. A interface de um objeto é seu lado público. O objeto usa seu lado privado, implementação, para mandar mensagens para a interface de outro objeto. A implementação de um objeto é o seu comportamento. Parte do seu comportamento é enviar mensagens para outros objetos que têm uma interface pública [OMG98a].

Os padrões de programação orientada a objetos se focalizam em dois aspectos principais da tecnologia de objetos:

- Linguagem de definição de interface (LDI)- que esconde as particularidades da implementação. O compilador usa estas definições de interface para gerar *stubs* que executam alguns serviços como checagem de tipos e manipulação de parâmetros;
- Serviços de sistema- são serviços básicos que simplificam as tarefas de programação. Por exemplo: serviços de nome provêm um diretório para converter nomes de objetos em endereços do sistema, serviços de notificação de eventos que enviam e recebem eventos entre objetos que executam vários papéis como consumidor, fornecedor, canais. O serviço de ciclo de vida cria, copia e remove os objetos.

O serviço de um *object broker* consiste em implementar a interação entre os objetos. O *broker* localiza os objetos e envia pedidos para eles. Da mesma forma como a programação orientada a objetos introduziu abstração importante para a representação das informações e seu comportamento, o conceito de *object broker* introduziu os mesmos tipos de conceitos para o gerenciamento de processos [OMG98a].

Linguagens de Programação

Uma gama enorme de linguagens de programação está disponível no mercado. Desde as linguagens puramente comerciais como COBOL até as linguagens de terceira e quarta geração, linguagens visuais e as mais novas linguagens Orientadas a Objeto. Cada uma tem seus pontos fortes e fracos em diferentes níveis de abstração. Os requisitos do novo

sistema que deve ser construído e a arquitetura escolhida para a organização irão apontar a melhor linguagem a ser utilizada. Exemplo de fatores que devem ser analisados, nos mais diferentes níveis de abstração:

- facilidade de programação;
- desempenho;
- depuradores;
- compiladores;
- tipos de dados;
- geração de bibliotecas dinâmicas;
- abstração;
- facilidade de reuso.

Algumas se tornaram padrão como o SQL e outras têm demonstrado grande poder de abstração da plataforma *hardware* que executam como por exemplo a linguagem Java ([SM01]).

O último passo desta fase é a definição de quais sistemas de informação serão adquiridos baseado nas opções do mercado que preencham todos os requisitos da arquitetura de sistemas de informação da companhia ou se haverá a necessidade de implementação caso os sistemas comerciais não preencham os requisitos da companhia.

3.4 Migração Incremental

Após ter sido feito, nas fases anteriores, o mapeamento dos processos de negócio da empresa, o relacionamento dos sistemas de informações legados com os processos de negócio, a priorização de migração dos sistemas legados e definida a arquitetura de sistemas de informação da organização, deve-se iniciar a construção do novo sistema e a migração propriamente dita dos sistemas ou subsistemas legados.

Os passos que serão apresentados a seguir nesta última fase da metodologia estão baseados nas propostas apresentadas por Brodie et al. [BS95], Ganti et al. [GB95] e Bergey et al. [BNS97]. O processo de validação da metodologia também trouxe realimentações para a definição das etapas desta fase.

Esta fase da metodologia está dividida em quatro grande subfases. A seguir serão mostrados os objetivos e abordagens de cada uma das subfases.

1. Detalhar o sistema legado

Objetivos

- identificar as partes do sistema legado e classificá-las como não estruturado, semi-estruturado e estruturado;
- identificar as regras de negócio dentro do sistema legado;
- identificar os dados importantes e os que serviam como apoio para os sistemas legados e que devem ser descartados;
- identificar dados de sistema e dados de configuração e parametrização;
- classificar os dados quanto à sua propriedade (usuários e sistemas) e autonomia (público ou privado);
- classificar os dados quanto ao seu tipo: transacional ou suporte a decisão.

Abordagem

- detalhar os processos de negócio que envolvem o sistema legado;
- levantar toda a documentação possível relativa ao sistema legado e sistemas relacionados;
- levantar todas as telas do sistema;
- realizar entrevistas com os usuários;
- realizar entrevistas com os desenvolvedores;
- utilizar alguma metodologia e ferramenta para mapeamento e classificação dos dados.

2. Projetar e construir o sistema de informação destino

Objetivos

- Projetar e construir as interfaces de usuário do novo sistema;
- Projetar e construir as interfaces entre o novo sistema e os sistemas legados e outros sistemas existentes na companhia;
- Projetar e construir os bancos de dados transacionais e de suporte a decisão (*data warehouse*);
- Projetar e construir os componentes que implementam as regras de negócio.

Abordagem

- utilizar o mapeamento detalhado dos processos de negócio;

- utilizar os resultados do detalhamento do sistema legado;
- seguir os requisitos identificados durante a elaboração da arquitetura global de sistemas de informação da companhia;
- utilizar alguma metodologia de garantia da qualidade do processo e do produto, como por exemplo o *Capability Maturity Model (CMM)* [PCC93] do *Software Engineering Institute (SEI)*;
- utilizar uma metodologia e ferramentas para projeto de sistemas de informação, como por exemplo a metodologia orientada a objetos;
- utilizar ambientes de desenvolvimento para agilizar a fase de codificação do novo sistema de informação;
- utilizar metodologias e ferramentas de teste de sistemas;
- utilizar as plataformas de *hardware* e *software* escolhidos na definição da arquitetura global de sistemas de informação durante a construção do novo sistema de informação.

3. Projetar e construir ferramentas para conversão dos dados do sistema legado para o sistema destino

Objetivos

- Projetar e construir ferramentas para exportação dos dados legados para um formato canônico;
- Projetar e construir ferramentas para importação de dados a partir de um formato canônico;
- Projetar e construir ferramentas de análise da quantidade e qualidade dos dados importados.

Abordagem

- utilizar o mapeamento detalhado dos processos de negócio;
- utilizar os resultados do detalhamento do sistema legado;
- utilizar os resultados do projeto do novo sistema de informação;
- utilizar uma metodologia e ferramentas para projeto de sistemas de informação, como por exemplo a metodologia orientada a objetos;
- utilizar ambientes de desenvolvimento para agilizar a fase de codificação do novo sistema de informação;
- utilizar metodologias e ferramentas de teste de sistemas.

4. Definir o plano de implantação do ambiente final e migração do sistema legado

Objetivos

- garantir que todos os detalhes necessários para a correta instalação e configuração do novo sistema estejam satisfeitas;
- garantir que todos os pré-requisitos de *hardware* e *software* estarão instalados e testados quando o novo sistema estiver pronto;
- garantir que todos os dados dos sistemas legados serão corretamente migrados para o novo sistema;
- garantir que o novo sistema entrará em produção da forma mais correta possível;
- garantir que todas as pessoas-chaves no processo estejam presentes durante a fase de migração do sistema legado.

Abordagem

- utilizar o mapeamento detalhado dos processos de negócio;
- utilizar os resultados do detalhamento do sistema legado;
- utilizar o documento de arquitetura de sistemas de informação da companhia;
- utilizar os documentos de projeto do novo sistemas de informação;
- elaborar um plano de implantação para o novo sistema que está sendo elaborado;
- acompanhar a instalação e configuração dos componentes de *hardware* e *software* necessários para o novo sistema;
- realizar o treinamento dos usuários finais no novo sistema;
- realizar migrações preliminares dos dados do sistema legado para o novo sistema para testar o processo, validar as ferramentas e corrigir os dados do sistema legado;
- realizar testes piloto do novo sistema;
- colocar o novo sistema em produção com o sistema legado executando em paralelo, quando possível, garantindo eventuais problemas não identificados até o momento. Este estado deve permanecer por um tempo curto e determinado pois os efeitos de uma volta do sistema legado em produção não foram detalhadas e podem ser desastrosos para a consistência dos dados.

Esta é a fase mais complexa e longa de toda a migração. Pela quantidade de tarefas mapeadas pode-se notar que a equipe envolvida nesta fase é muito grande e com os mais variados perfis como: arquitetos de sistemas, administradores de dados, administradores de banco de dados, especialistas em integração de sistemas, analistas de sistemas, analistas de suporte e programadores. Para que todas estas atividades possam ser executadas de forma correta e dentro do cronograma e orçamento do projeto o esforço de gerenciamento do projeto e das equipes também é muito grande.

3.5 Resumo do Capítulo

Neste capítulo foram detalhadas as fases da metodologia de migração dos sistemas legados para novos sistemas de informação mais flexíveis às mudanças dos processos de negócio que as grandes empresas vêm passando. A metodologia possui três grandes fases e vários passos dentro de cada uma.

Na fase inicial é feito um levantamento geral da situação dos processos e sistemas de informação existentes na empresa. Na fase seguinte é feita uma análise da arquitetura dos sistemas legados e uma proposta de arquitetura para todos os sistemas de informação da empresa, sejam eles legados ou não, de forma a permitir a interoperabilidade entre os mesmos. Na fase final é realizada a migração do sistema legado, sendo esta a fase mais longa e mais difícil por causa dos riscos envolvidos. A cada migração de um sistema legado ou parte dele as duas primeiras fases podem ser realimentadas com relação a requisitos e restrições não identificados anteriormente.

A seguir são apresentados os passos de cada fase da metodologia:

1. Análise global
 - (a) Mapeamento dos processos do negócio;
 - (b) Mapeamento dos sistemas de informação em relação aos processos do negócio;
 - (c) Análise dos processos de negócio e sistemas legados;
 - (d) Priorização dos processos e sistemas legados a serem migrados.
2. Definição de uma arquitetura para os sistemas de informação
 - (a) Mapeamento da arquitetura dos sistemas de informação existentes;
 - (b) Definição da nova arquitetura de sistemas de informação.
3. Migração incremental dos sistemas legados
 - (a) Detalhar o sistema legado;

- (b) Projetar e construir o sistema de informação destino;
- (c) Projetar e construir o ferramentas para conversão dos dados do sistema legado para o sistema destino;
- (d) Definir o plano de implantação do ambiente final e migração do sistema legado.

Capítulo 4

Estudo de Caso da Metodologia

Neste capítulo será detalhada como ocorreu a validação da metodologia proposta nesta dissertação durante a aplicação da mesma em um caso real. O estudo de caso utilizado será o SAGRE (Sistema Automatizado para Gerenciamento da Rede Externa) [Ma97].

Durante a implantação do SAGRE em uma Empresa Operadora de Telecomunicações (EOT) foi necessária a adaptação dele para atender as necessidades particulares da empresa, migração de parte dos dados dos sistemas legados para o SAGRE, instalação de mediadores de dados ou aplicação e finalmente o *cut-over* do sistema legado. Nesta implantação a metodologia pôde ser validada e realimentada com necessidades que surgiram ao longo desta implantação.

A seguir será feita uma descrição do SAGRE, uma breve explicação de alguns sistemas legados existentes na EOT. Depois será aplicada a metodologia proposta no capítulo 3 em um caso real ocorrido na implantação do SAGRE e por fim será feito um resumo do capítulo.

4.1 Projeto SAGRE

O SAGRE foi desenvolvido para atender as necessidades das EOT de ferramentas automatizadas para converter os seus cadastros de rede externa de telecomunicações dos atuais mapas em papel para meio magnético e posteriormente gerenciar os processos de planejamento, projeto, construção, operação e manutenção da rede externa. Os módulos que compõem o sistema são:

- Conversão - ferramentas que auxiliam na conversão dos mapas em papéis para o SAGRE;
- Cadastro - faz a manutenção dos dados já convertidos através de atualizações diretas ou através de projetos, utilizando um esquema de versionamento dos dados

[DGMa95];

- Projeto - subsistema que possui ferramentas para edição, análise, cálculo de material e mão-de-obra e orçamento de projeto [GEOG94], [PTO95];
- Planejamento - conjunto de ferramentas para análise da rede utilizando um subconjunto dos elementos de rede;
- Operação - faz a gerência das movimentações de facilidades¹ de rede externa utilizadas na oferta de serviços de telecomunicações;
- Mercado - ferramentas para análise dos mercados atuais e futuros para exploração dos serviços de telecomunicações.

Desta forma o projeto consegue cobrir todo o ciclo de vida da rede externa, desde o seu planejamento até a sua operação. A característica mais importante do produto é a integração no mesmo sistema dos ambientes de projeto e operação da rede externa de telecomunicações [Ma00].

Devido à natureza espacial do SAGRE, é utilizado um Sistema de Informações Geográficas (SIG) como componente da solução. Todos os dados georeferenciados da rede externa estão digitalizados sob o Mapeamento Urbano Básico² (MUB). Enquanto a maioria dos SIG comerciais utilizam estruturas especiais e proprietárias para o armazenamento dos dados espaciais, o SIG *Vision* [Inc98], utilizado pelo SAGRE, apresenta uma característica diferente dos demais sistemas da sua categoria por utilizar apenas o SGBD para armazenagem tanto os dados gráficos como os dados alfanuméricos.

Esta abordagem pode tornar o SIG menos eficiente na recuperação dos dados espaciais em relação a outros SIG comerciais, devido à utilização das estruturas de dados convencionais do SGBD para implementar as estruturas de dados e índices espaciais [Car98, FMa92]. No entanto os demais SIG comerciais que utilizam estruturas proprietárias para o armazenamento dos dados espaciais, não preenchem todos os requisitos necessários para uma solução corporativa e georeferenciada, como por exemplo: segurança, confiabilidade, escalabilidade e base de dados contínua.

Uma nova tendência tem surgido no mercado de SIG, a utilização dos novos recursos dos SGBD comerciais para as estruturas espaciais [Cor01, Yar97]. Esta solução parece ser bastante promissora para o de armazenamento e recuperação de informações porém os representantes de SIG terão que adaptar seus produtos para utilizar os recursos nativos dos SGBD ou os fabricantes de SGBD terão que fornecer ferramentas para edição e visualização de dados espaciais.

¹equipamentos, cabos e fios necessários para o atendimento de um cliente de telecomunicações

²conjunto de objetos georeferenciados que compõem os mapas das cidades

4.2 Sistemas Legados na EOT

As EOT são empresas que apresentam um grande número de sistemas legados devido a vários fatores: elas foram as primeiras empresas a utilizar computadores e sistemas de informação, necessidade de manipular grande quantidade de informações e por terem muitos recursos para investimento.

A EOT alvo deste estudo possui três grandes sistemas legados: Sistemas de Gerenciamento de Clientes (SGC)³, Sistemas de Gerência de Facilidades (SGF)⁴ e Sistema de Demanda Telefônica (SDT)⁵. Estes sistemas começaram pequenos, há cerca de 20 anos, com apenas algumas funcionalidades de armazenamento dos dados dos clientes. Posteriormente, foram sendo expandidos para guardar dados de contratos, informações sobre os equipamentos das centrais telefônicas e por fim das facilidades de rede utilizadas pelos clientes.

Seu escopo foi sendo transformado de sistemas cadastrais *off-line* para sistemas de gerenciamento de clientes *on-line*. O volume de dados armazenados juntamente com o número de transações tem aumentado na ordem de 20% a 30% ao ano. Desta forma esses sistemas tornaram-se sistemas de missão crítica [PMa02] para as EOT, e com um volume de dados da ordem de Gigabytes.

Além disto, são sistemas híbridos [BS95], com pouca ou nenhuma documentação e rígidos, isto é, sem agilidade para mudanças. Eles estão escritos em linguagem COBOL e NATURAL, executam em *mainframes* e utilizam, na maioria dos casos, o SGBD ADABAS.

Estes sistemas não são escaláveis e suportam o volume de transações atual devido à capacidade de processamento das máquinas que estão sendo utilizadas. Porém esta solução tem-se mostrado muito cara e favorecendo a desativação dos sistemas.

Apesar dos problemas citados, esses sistemas apresentam um alto grau de confiabilidade pois a maioria dos erros já foram corrigidos além de executarem 24 horas por dia e 7 dias por semana e com uma carga de trabalho da ordem de milhares de transações por dia. Isto torna o processo de migração ainda mais complicado por aumentar a importância do requisito de confiabilidade do novo sistema.

Diante das características apresentadas como: rigidez do sistema, falta de documentação, grandes custos de manutenção, não escalável e natureza de missão crítica em que o sistema está inserido; é evidente que uma migração do tipo *Cold Turkey* [BS95] seria inviável. A seguir serão apresentados os passos da metodologia aplicados durante a migração destes sistemas.

³SGC - sistema que faz o tratamento de todas as solicitações dos clientes

⁴SGF - sistema que controla a alocação e estado das facilidades da rede de telecomunicações

⁵SDT - sistema responsável pela coleta e análise das necessidades a curto, médio e longo prazo de serviços de telecomunicações por parte dos usuários atuais e futuros

4.3 Análise Global

Nesta seção serão analisados apenas os processos ligados à área de engenharia e de atendimento a clientes das EOT pois são áreas onde o SAGRE atua. Os processos ligados as áreas de administração, jurídica, financeira e vendas não foram mapeados.

4.3.1 Mapeamento dos Processos

A seguir serão apresentados os atores, modelos de caso de uso (figuras 4.1 e 4.2) e os modelos de objeto (figuras 4.3, 4.4, 4.6 e 4.5) que representam os processos nas áreas de engenharia e operação.

Atores

Área de Gerenciamento a Clientes: área de empresa responsável por prover o gerenciamento aos clientes.

Área de Mercado: área da empresa responsável por fazer o levantamento e monitoração do mercado de telecomunicações.

Área de Planejamento: área responsável por fazer o planejamento da rede de telecomunicações.

Área de Projeto: área de empresa responsável por projetar e/ou inspecionar a execução dos projetos de rede externa.

Cliente: qualquer indivíduo ou empresa que tenha contratado algum tipo de serviço de telecomunicações.

Instalador: indivíduo encarregado por executar os serviços de construção/reparo na rede de telecomunicações.

Cliente Potencial: qualquer indivíduo ou empresa que possivelmente contrate algum tipo de serviço de telecomunicações.

Terceirizado: empresas prestadora de serviços de projeto e/ou construção de rede externa.

Caso de Uso LEVANTAMENTO DE MERCADO

Pesquisa de Mercado: este fluxo descreve como é feito uma pesquisa de mercado, como os dados são coletados e armazenados para posterior análise.

Análise de Mercado: este fluxo utiliza os dados coletados durante a pesquisa de mercado e através de cálculos, simulações e projeções faz uma avaliação e análise de tendências dos segmentos de mercado.

Planos de Mercado: este fluxo traça as metas de gerenciamento dos segmentos de mercado. Neste plano devem estar as diretivas para tipos de atendimento, isto é, tecnologias

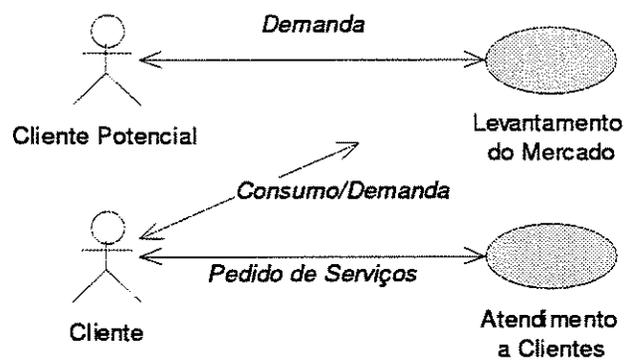


Figura 4.1: Casos de Uso de Atendimento a Clientes

que devem estar disponíveis para cada segmento do mercado. Este plano deve considerar o orçamento, planos estratégicos e planos de investimento definidos pela empresa. Ele deve ser elaborado anualmente e revisto sempre que novas tecnologias e serviços estiverem disponíveis ou as metas da empresa não estiverem sendo atingidas.

Caso de Uso ATENDIMENTO A CLIENTE

Pedido de serviços: este fluxo é responsável por coletar os dados do cliente e coletar o tipo de serviço que ele está requisitando. Os tipos de serviços variam de acordo com o tipo do cliente e a região em que ele se encontra. Por exemplo, em locais onde a rede ainda não está digitalizada, serviços de secretária eletrônica, *follow me*, etc, não estão disponíveis.

Alguns tipos de atendimento são executados imediatamente como serviços de auxílio à lista. Outros, porém, requerem uma investigação para verificar a viabilidade técnica e o prazo para disponibilização do mesmo. Para estes serviços existe um documento interno às EOT chamado de Ordem de Serviço (OS). Nele são armazenadas todas as informações sobre a sequência de atendimento do cliente, desde o seu pedido até o desfecho do atendimento.

Comunicação de defeito: este fluxo é responsável por coletar os dados do cliente e o tipo de defeito que ele está informando. Os defeitos são investigados e posteriormente é retornado ao cliente o motivo do defeito, caso o defeito realmente exista, e as ações que foram tomadas para corrigi-lo. Todos estes detalhes são armazenados em um documento chamado Bilhete de Defeito (BD).

Geração de Ordem de Serviço (OS): este é o principal documento da área de gerenciamento a clientes. Nele estão descritas as operações necessárias para o atendimento do cliente e os prazos de atendimento. Este documento é criado e finalizado pela área de gerenciamento a clientes mas é o instrumento para que todas as áreas de empresa envolvidas no atendimento do cliente, possam estar cientes e executarem as atividades que lhes são pertinentes. Este tipo de documento serve como base para métricas de atendimento e controle de qualidade da rede.

Geração de Bilhete de Defeito (BD): este documento descreve as operações necessárias para a verificação e eliminação do problema relatado pelo cliente ou usuário de serviços de telecomunicações. Assim como as OS, os BD podem ser encaminhados para outras áreas dentro da EOT até que ele possa ser classificado como concluído pela área de gerenciamento a cliente. Este tipo de documento serve como base para métricas de defeito e controle de qualidade da rede. Também é utilizado para gerar relatórios de indicadores de defeitos para o órgão regulador governamental.

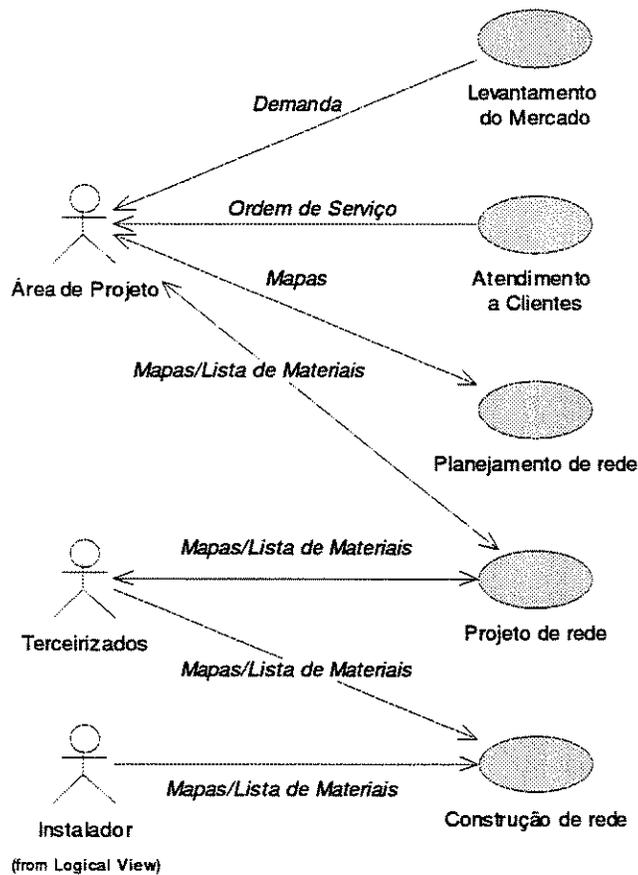


Figura 4.2: Casos de Uso de Engenharia

Caso de Uso PLANEJAMENTO DE REDE

Análise da rede atual: este fluxo é responsável pela avaliação da rede atual de uma determinada região. Esta região pode ser um estado, uma cidade, uma área de estação. Devem ser levados em consideração os macro-elementos da rede de telecomunicações para um estudo de alto nível e as possibilidades de expansão e modernização da rede.

Esta área da empresa é responsável pelos estudos de médio e longo prazo. Os dados para estes estudos são a rede atual, os dados de pesquisa de mercado, o planejamento estratégico da empresa e as tendências tecnológicas na área de telecomunicações.

Proposta de expansão: Este documento é um dos resultados do planejamento. Ele pode ser textual ou mapas indicando locais de instalação de novos elementos de rede.

Proposta de modernização: Este é outro documento resultante da atividade de planejamento. Ele indica os elementos de rede que devem ser removidos, substituídos e os novos que devem ser inseridos para que a atual rede possa se modernizar, o atendimento aos clientes possa ser melhorado e para que a EOT possa fazer frente às empresas concorrentes.

Caso de Uso PROJETO DE REDE

Análise das OS: para alguns tipos de atendimento aos clientes pode ser necessário a realização de pequenas alterações na rede de telecomunicações. Os pedidos e a rede atual devem ser analisados e um novo projeto deve ser proposto.

Pingação da demanda: este fluxo deve ser feito para que se possa iniciar grandes projetos de expansão da rede de telecomunicações. Esta atividade consiste em colocar as demandas por serviços de telecomunicações para os próximos anos em cada imóvel e sua totalização por eixos de rua ou faces de quadra. Estes dados são obtidos a partir das pesquisas de mercado e das projeções realizadas pela área de Mercado.

Estudo de Alternativas: tanto em projetos de expansão como em projetos de atendimentos rápidos a clientes são elaboradas algumas alternativas de projeto para que seja escolhida a mais viável técnica e economicamente.

Lista de Materiais: neste fluxo é feita a lista dos materiais necessários para que o projeto possa ser executado. Esta lista é baseada nos padrões de materiais e fornecedores previamente adotados pela EOT.

Lista de mão-de-obra: neste fluxo são definidos os tipos e quantidade de mão-de-obra necessária para a execução do projeto. A unidade de medida é homens/hora.

Orçamento: baseado na lista de materiais e mão-de-obra é feito o orçamento do projeto.

Seqüência de serviços: após escolhida a alternativa de projeto a ser executada e dependendo do tamanho do projeto, este pode ser dividido em subprojetos. É feito um diagrama

de PERT para estabelecer a ordem e dependências de execução de cada subprojeto.

Caso de Uso CONSTRUÇÃO DE REDE

Verificação de estoque: com a lista de materiais em mãos, o instalador verifica aqueles que estão disponíveis em estoque e a sua quantidade.

Pedido de compra: para os materiais que não existem em estoque ou possuem quantidades inferiores às necessárias, é feito um pedido de compra.

Contratação de mão-de-obra: neste fluxo a área de construção de rede faz a contratação da mão-de-obra necessária para a execução do projeto.

Instalação da rede: neste fluxo, com os materiais disponíveis e a mão-de-obra contratada, é feito a construção e a instalação da nova rede. Nos casos de alteração de rede já existente, deve-se tomar todos os cuidados necessários para que os atuais usuários de serviços de telecomunicações sejam interrompidos o menor tempo possível, durante as instalações.

Orçamento real: após a construção da nova rede é feita a revisão do orçamento gerado pelos projetistas.

Liberação da nova rede: após a construção e testes, a nova rede é liberada para a área de gerenciamento a clientes para que as novas facilidades de rede e os novos serviços disponíveis possam ser comercializados.

Objetos de Interface

Atendente: responsável por fazer o atendimento aos clientes, usuários e potenciais clientes de serviços de telecomunicações.

Gerente de Instalação: responsável por acompanhar e monitorar a instalação do projeto de rede externa.

Gerente de Projeto: responsável por acompanhar e aprovar o projeto de rede externa.

Pesquisador de Mercado: faz o levantamento do mercado, classificando-o segundo critérios pré-estabelecidos.

Analista de Mercado: faz a análise e reclassificação, quando necessário, do mercado levantado.

Gerente de Planejamento: responsável por acompanhar o planejamento da rede de telecomunicações para o atendimento da demanda para os próximos anos.

Objetos de Controle

Gerenciador de Estoque: responsável por fazer o controle dos itens existentes no estoque e a reposição dos mesmos.

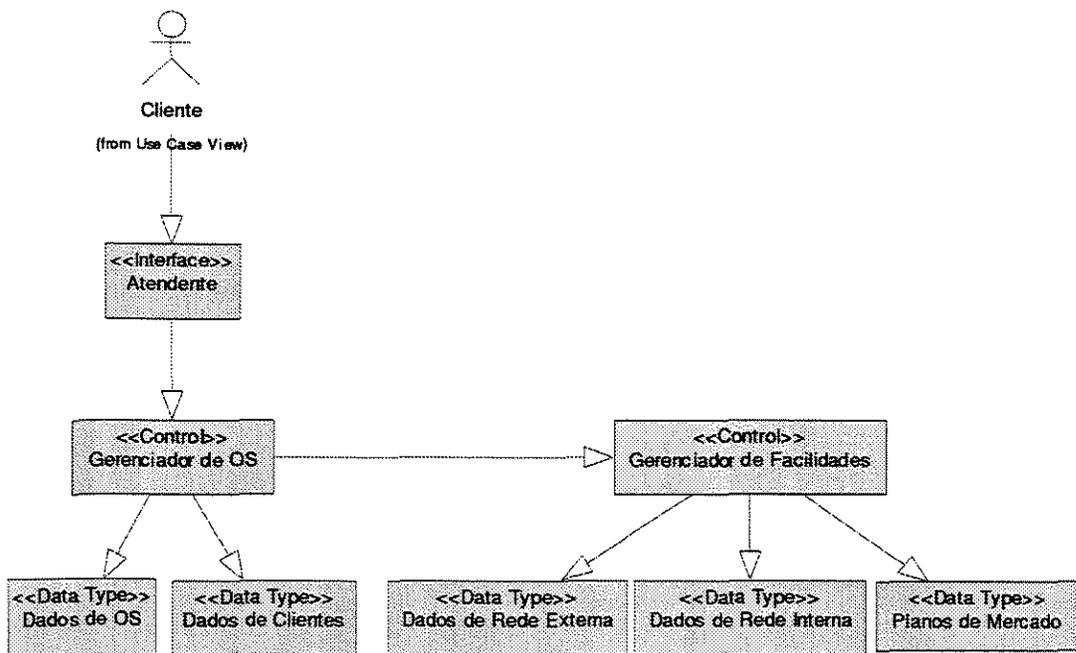


Figura 4.3: Modelo de Objetos de Atendimento a Clientes

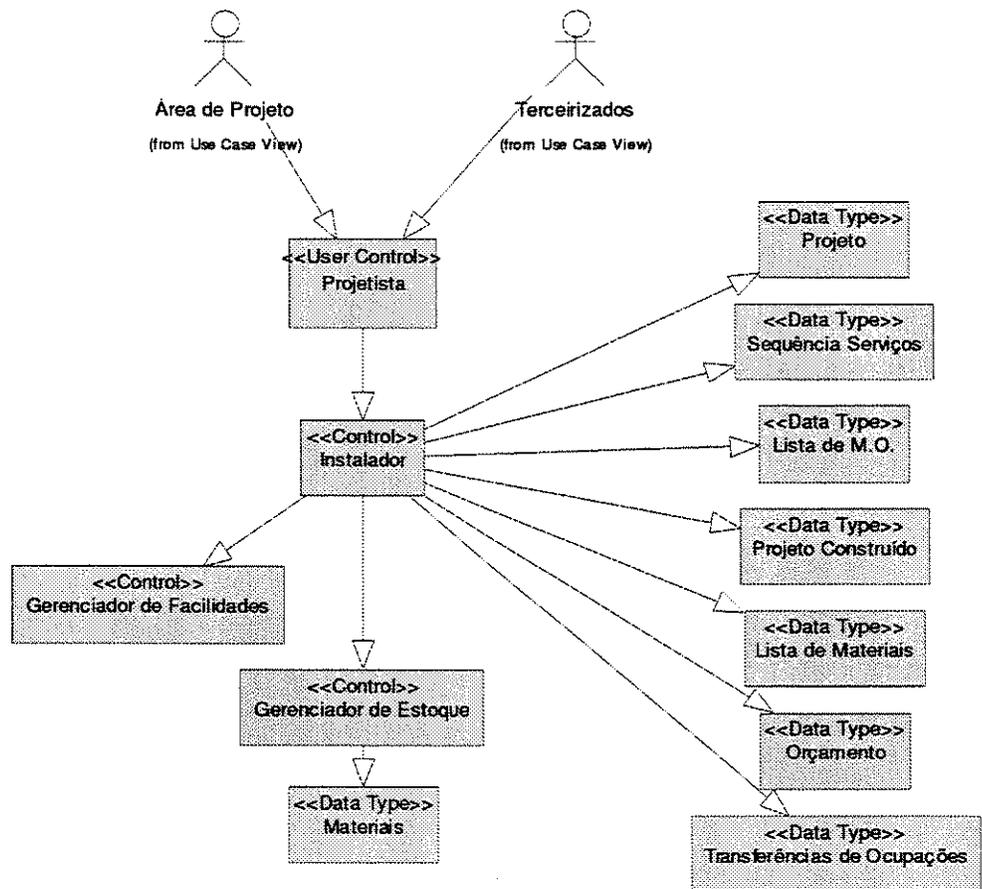


Figura 4.4: Modelo de Objetos de Construção de Rede

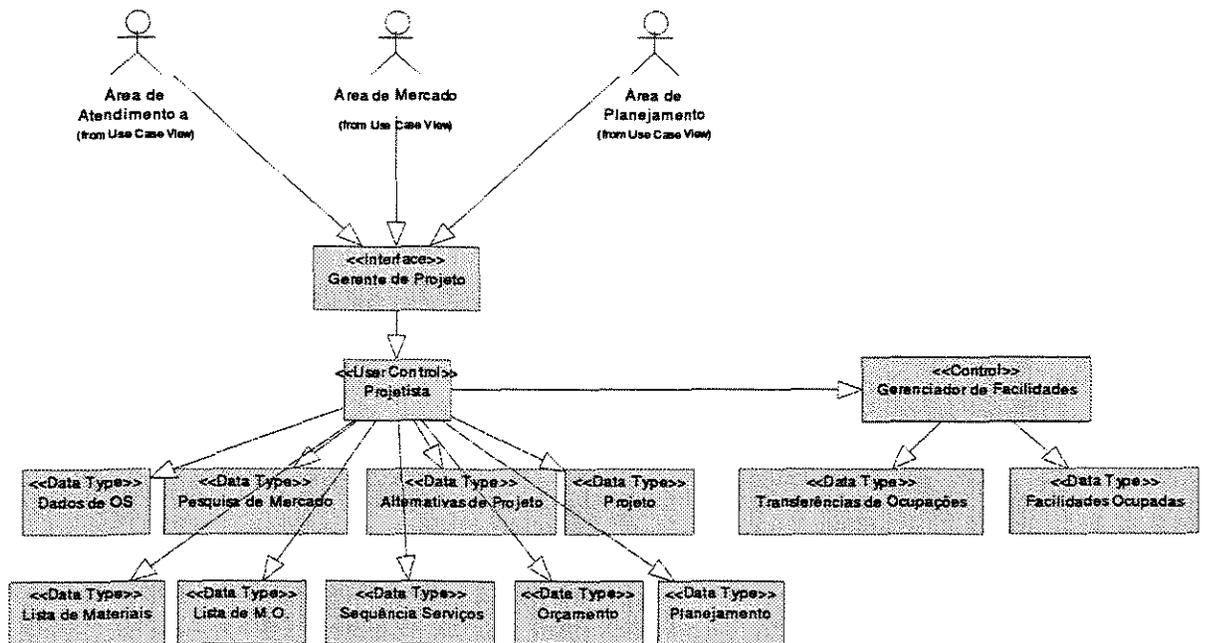


Figura 4.5: Modelo de Objetos de Projeto

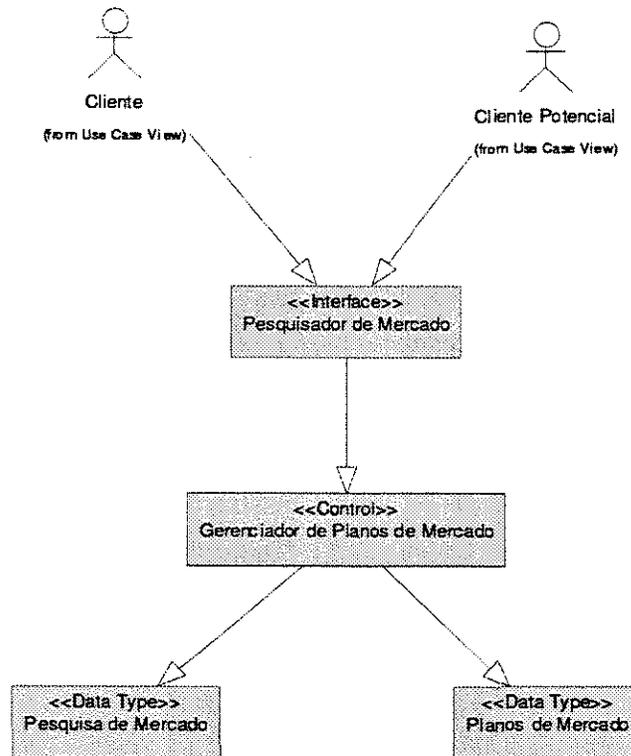


Figura 4.6: Modelo de Objetos de Mercado

Gerenciador de Facilidades: faz o gerenciamento das facilidades que estão livres, ocupadas, reservadas e com defeito.

Gerenciador de OS: controla o fluxo de uma OS desde a sua abertura até a sua resolução (fechamento).

Gerenciador de Planos de Mercado: faz os planos de mercado de acordo com a pesquisa e análise de mercado.

Instalador: responsável por fazer a instalação dos elementos de rede externa que foram projetados.

Projetista: responsável por fazer os projetos de acordo com as análises de mercado, planejamento de rede e demanda por serviços de telecomunicações dos clientes.

Objetos de Entidade

Alternativas de Projeto: representa as várias possibilidades de projeto para uma determinada situação de atendimento ao cliente.

Contrato: informações sobre o cliente, tipo de serviço adquirido e tempo de utilização do mesmo.

Dados de Clientes: informações sobre os clientes.

Dados de OS: informações de uma OS como: cliente, data de abertura, motivo, situação atual da OS, data de fechamento, etc.

Dados de Rede Externa: informações sobre as infra-estrutura de telecomunicações localizada fora dos prédios das EOT.

Dados de Rede Interna: informações sobre as infra-estrutura de telecomunicações localizada dentro dos prédios das EOT.

Facilidades Ocupadas: elementos de rede interna e externa alocados para um determinado cliente/serviço.

Lista de Materiais: informações de materiais como: tipo, fabricante, necessárias para a instalação do projeto de rede externa.

Lista de M.O.: lista de tipo de mão-de-obra necessária para a instalação do projeto de rede externa.

Materiais: dados de materiais.

Orçamento: informações sobre o custo do projeto baseado na quantidade de material e mão-de-obra necessários.

Pesquisa de Mercado: dados do mercado atual e futuro para serviços de telecomunicações por endereço ou por região.

Planejamento: anteprojetos da rede externa de telecomunicações.

Planos de Mercado: representa as estratégias de mercado para atingir novos clientes.

Projeto: informações de remoção, redistribuição, instalação e ampliação de elementos de rede externa.

Projeto Construído: dados dos elementos de rede externa que realmente estão em campo ao final da construção do projeto.

Sequência de Serviços: sequência de construção dos elementos de rede para que os serviços atuais possam continuar operando e os novos possam ser disponibilizados rapidamente.

Serviços disponíveis: conjunto de serviços de telecomunicações disponíveis para os clientes e clientes potenciais.

Tipos de Serviço: classificação dos serviços de telecomunicações.

Transferência de Ocupações: informações sobre os velhos e novos elementos de rede utilizados pelos clientes.

	Atendimento Cliente	Mercado	Planejamento de rede	Projeto de rede	Construção de rede
Sistema de Geren. Clientes	X			X	X
Sistema de Designção Facilidade	X			X	X
Sistema de Demanda Telef.		X	X	X	
Sistema de Materiais e MO				X	X

Tabela 4.1: Matriz de Mapeamento Processos X Sistemas da EOT

4.3.2 Mapeamento dos Sistemas de Informações

Nesta fase da metodologia foi construída uma tabela (tab. 4.1) para fazer o mapeamento entre os sistemas de informação existentes e os casos de uso em que eles atuavam. A tabela auxiliou no processo de tomada de decisão da ordem de migração dos sistemas legados pois ela foi baseada em critérios técnicos, estratégicos e financeiros.

4.3.3 Análise dos Processos

Normalmente antes da introdução do SAGRE nas EOT, os projetos de rede externa eram feitos em papel. Uma ou mais visitas ao campo eram feitas para verificar se o cadastro em papel estava igual à rede existente em campo e só então o projetista fazia o projeto sobre a planta em papel. Após a conclusão e aprovação do projeto ele era construído em campo, o instalador fazia as alterações necessárias devido às alterações encontradas em campo e o projeto real era devolvido para atualizar o cadastro da rede em papel.

A única ferramenta automatizada de auxílio ao projetista na EOT eram os Sistemas de Materiais e Mão-de-obra. Porém estes sistemas eram muito simples, servindo praticamente como planilhas para fazer a totalização dos materiais e mão-de-obra necessárias para o projeto, além de obrigar o projetista a entrar com todas as informações que já haviam sido colocadas durante o projeto em papel.

Quando os projetos eram terceirizados, eles eram feitos utilizando-se ferramentas CAD, porém sem um controle de escala ou de acurácia no georeferenciamento e sem a preocupação de formar uma base de dados única para todas as plantas. Em outras palavras, projetos distintos que eram feitos sob a mesma área, não utilizavam as mesmas plantas

digitalizadas, resultando em plantas distintas para a mesma área e impossível de serem unidas [PTO95].

Podemos concluir então que um sistema para automatizar os processos de projeto e construção de rede era muito importante para a EOT em estudo.

Já os processos de classificação de Mercado possuem uma ferramenta para cadastro das informações censitárias colhidas em campo, correções de erros de levantamento em campo ou digitalização de dados e ferramentas de cálculo de demanda. O resultado deste processo é um banco de dados com informações para auxiliar a tomada de decisão de ampliação e modernização de rede que será feito durante a fase de planejamento. Estes sistemas apesar de se enquadrarem no conceito de sistemas legados, estão preenchendo as expectativas das EOT e também não são sistemas de missão crítica [PMa02] nem são usados diretamente nos processos de atendimento a clientes.

Existe uma única necessidade premente nos sistemas de Mercado que não pode ser atendida pelo sistema atual: a espacialização das informações resultantes da coleta, correção e processamento dos dados para uma melhor análise e uma correlação com as informações de rede externa.

Os processos de classificação e análise de Mercado estão sendo reavaliado pela EOT pois a antiga forma de coleta de dados e alimentação do sistema legado requer um volume e qualidade muito grande de dados que são difíceis de serem coletados. A tendência atual é de se conseguir dados de fontes externas e fazer algumas transformações para que ele possa refletir a necessidade de serviços de telecomunicações por área de abrangência. Desta forma uma migração deste sistema para uma nova plataforma não traria um grande ganho para a EOT uma vez que a coleta de dados é o grande problema nestes processos.

A enorme competição existente hoje no mercado de telecomunicações tornou os processos de atendimento a clientes os mais importantes para qualquer empresa. Atualmente a EOT possui um sistema legado para gerenciamento dos processos de clientes. Este sistema está com enormes problemas para se adaptar aos novos serviços de telecomunicações que foram criados recentemente, agilidade em disponibilizar as novas redes para o público, falta de integridade entre o cadastro e a rede existente em campo e também com problemas de escalabilidade diante do grande aumento no volume dos clientes e conseqüentemente das solicitações.

Os processos de planejamento de rede até alguns anos atrás não tinham sentido pois com a enorme demanda por serviços de telecomunicações existentes no país, todo tipo de expansão e modernização das redes eram extremamente necessárias. Também pela carência na oferta de serviços básicos de telecomunicações, os novos serviços como *internet*, nem eram cogitados. A área de planejamento atuava mais como uma área normativa dentro da empresa. Atualmente, com o problema da demanda bem mais atenuado, esta área voltou a ter a função original que é prever possíveis crescimentos por serviços de

telecomunicações em novas áreas ou em áreas existentes, normalmente associada a novos serviços. Não existia um único sistema para automatizar estes processos de negócio na EOT. Na realidade existiam alguns esforços isolados de algumas áreas em elaborar pequenos bancos de dados, uso de planilhas, etc. Todas estas iniciativas eram isoladas e departamentais.

4.3.4 **Priorização dos Processos e Sistemas Legados**

A maior necessidade dentro da EOT encontrava-se nos processos de Projeto e Construção de rede devido aos seguintes aspectos:

- os cadastros de rede, em papel, encontravam-se desatualizados;
- o número de projetos sendo elaborados e construídos era enorme, devido à necessidade de atender a demanda por serviços;
- atendimento e antecipação das metas colocadas pelo órgão regulador de Telecomunicações;
- falta de uma ferramenta corporativa para auxiliar a elaboração e construção do projeto e manutenção do cadastro de rede.

A segunda maior necessidade da EOT era no processo de Atendimento a Clientes. Existiam sistemas automatizados que auxiliavam este processo porém a maior dificuldade estava justamente na falta de atualização cadastral destes sistemas, causando um grande número de atendimentos inválidos que resultam em:

- grande número de idas a campo sem resultado, isto é, os elementos de rede já estavam ocupados ou com defeito e o cliente não podia ser atendido;
- clientes não eram atendidos quando os novos elementos de rede se tornavam disponíveis;
- insatisfação dos clientes;
- clientes ficavam sem serviço (*mudos*) durante a construção de um projeto em uma determinada área devido ao cadastro de ocupação de facilidades estarem desatualizado em relação à rede existente.

Diante das análises feitas em relação aos processos de negócio da área de atendimento a clientes e engenharia e do grau atual de automação dos processos, a priorização de melhoria dos processos e migração dos sistemas legados deveria seguir a seguinte sequência:

1. processos de projeto e construção de rede;
2. processos de atendimento a clientes;
3. processos de planejamento;
4. processos de mercado.

Os processos de projeto, construção de rede e planejamento não estavam automatizados, existiam apenas algumas planilhas que auxiliavam em cálculos como de material e mão-de-obra do projeto ou alguns bancos de dados isolados para coleta de dados para planejamento além do uso de ferramentas CAD para desenho dos projetos de rede sem nenhuma padronização. Estes processos não foram analisados neste trabalho pois não se considerou que existisse migração de sistemas legados nesta situação.

Os processos de análise de mercado não eram os de maior importância para as EOT, uma vez que o maior problema está na aquisição de dados de boa qualidade para as mais diversas cidades e regiões do país. Desta forma foi apenas sugerida uma integração entre o SAGRE e o Sistema de Demanda Telefônica (SDT), complementando a fase de análise com informações especializadas. Este sistema também não foi migrado pela EOT e não foi alvo de estudo mais detalhado neste trabalho.

Os processos de atendimento a clientes possuem um sistema legado, o SGC, para gerenciamento completo dos fluxos de atendimento a clientes, cadastro das informações dos clientes e gerência de facilidades de rede. Como exposto na seção anterior estes sistemas são vitais para as EOT no mercado competitivo atual. Estes processos e o SGC não foram detalhados totalmente pois o SAGRE não atua nos processos específicos de atendimento e cadastro informações dos clientes. Foram detalhadas apenas as regras de negócio necessárias para o melhor entendimento do Sistema de Gerência de Facilidades (SGF).

Os fluxos dos processos de gerência de facilidades da área de atendimento a clientes foram detalhados pois o sistema que fazia este controle, o SGF, foi migrado para o módulo de Operação do SAGRE (SAGRE/Oper).

4.4 Arquitetura de Sistemas de Informações para a EOT

Nesta seção será feita uma análise da arquitetura dos sistemas legados, a sua forma de organização e os possíveis mediadores a serem utilizados. Posteriormente será detalhada a arquitetura de sistemas de informação utilizada na EOT dos fluxos e sistemas que

interagem com o SAGRE e também a arquitetura do próprio SAGRE por ele ser o sistema que substituiu parte dos sistemas legados.

4.4.1 Arquitetura dos Sistemas Legados

O Sistema de Gerenciamento de Clientes (SGC) era responsável pelo atendimento aos clientes, alocação de números na central telefônica e tramitador de Ordens de Serviço (OS) e Bilhetes de Defeito (BD). Além disto, ele possui o cadastro geral dos clientes e endereços da empresa.

Uma típica situação do Sistema de Gerenciamento de Clientes é o atendimento de um pedido de novo telefone. O SGC deveria executar os seguintes passos:

- Verificar se o endereço é um endereço válido;
- Verificar a disponibilidade de facilidades de central telefônica;
- Verificar a disponibilidade de facilidades de rede externa no endereço;
- Alocar todas as facilidades da rede externa e central telefônica;
- Informar o cliente do prazo de atendimento;
- Emitir OS para instalação em campo;
- Após a execução do serviço fazer a confirmação da alocação das facilidades da rede externa e central telefônica;
- Avisar o cliente da disponibilidade dos serviços;
- Enviar notificação de cliente ativo para o sistema de Bilhetagem e Faturamento;
- Dar baixa na OS.

Este era o fluxo sem nenhuma situação de contorno. Normalmente existem vários problemas em cada um dos fluxos e situações de contorno para cada um.

O SGC não fez parte do estudo detalhado pois ele não executa funções que sobreponham as funções do SAGRE, porém a sua parte de Gerência de OS e BD era extremamente importante para o estudo de caso pois ele fazia a interface dos pedidos dos clientes com o Sistema de Gerência de Facilidades (SGF) que foi substituído pelo módulo de Operação do SAGRE.

Da mesma forma que o SGC o SGF também apresentava uma arquitetura híbrida. Apesar da rigidez das arquiteturas, a integração entre os dois sistemas era transparente pois eles compartilhavam os mesmos *data files* do SGBD ADABAS, chegando ao ponto de

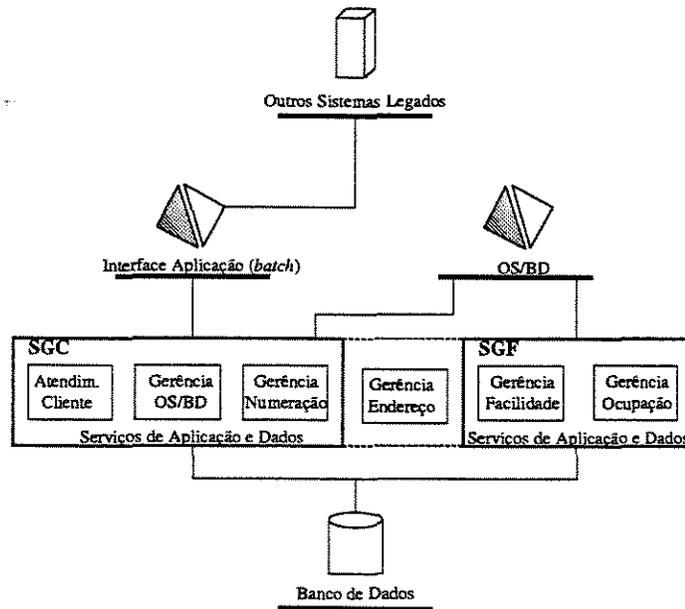


Figura 4.7: Arquitetura dos sistemas legados SGC e SGF

se confundirem quanto a um ou dois sistemas. Muitas vezes eles pareciam ser apenas um sistema com várias funções, outras vezes possuíam características distintas como: estilo e linguagem de programação distintas ou códigos duplicados. O único cuidado que existia era em relação às permissões de escrita e leitura nos *data files* de cada um dos sistemas.

A figura 4.7 detalha a arquitetura dos sistemas Gerência de Clientes e Gerência de Facilidades. Na figura fica clara a falta de organização dos sistemas. As funcionalidades de Gerência de Endereço eram utilizadas pelos dois sistemas sem nenhum controle. Note que neste caso não se tratava de reuso de código com todos os conceitos de encapsulamento e herança que prega a metodologia orientada a objetos e sim uma falta de organização e uma falta de coordenação dos limites de cada um dos sistemas.

4.4.2 Arquitetura dos novos Sistemas Informação

Nesta seção será apresentada a arquitetura dos novos sistemas de informação da empresa e a sua integração com os sistemas legados. Para que possam ser explorados seus detalhes a arquitetura proposta será apresentada em três níveis de abstração: Arquitetura de Rede,

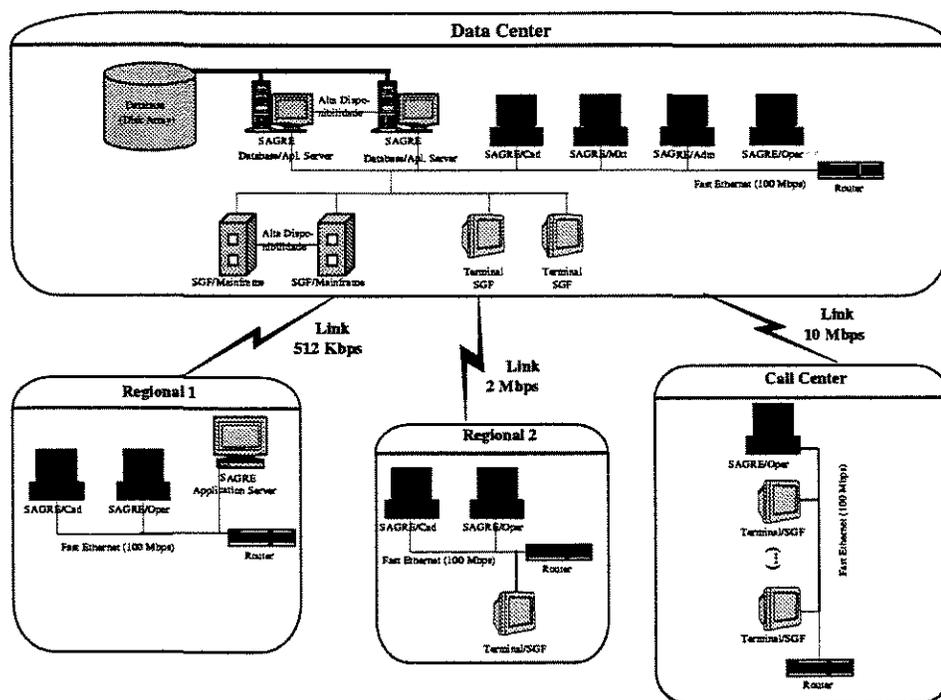


Figura 4.8: Arquitetura de Rede

Arquitetura de Integração de Sistemas e Arquitetura de Sistemas de Informação.

Arquitetura de Rede

No início da implantação dos módulos de Cadastro e Projeto de rede do SAGRE ficou clara a necessidade de uma remodelagem do processo de instalação do sistema e de uma maior padronização do ambiente operacional da EOT. Iniciou-se um estudo detalhado do *hardware* disponível, topologia de rede a ser utilizada, necessidades de distribuição e necessidades de banda. Ao final deste trabalho foi elaborado uma proposta de configuração de ambiente para a EOT. Este documento foi aprovado e implantado em toda a empresa.

A figura 4.8 detalha a arquitetura de rede proposta para a empresa.

Arquitetura de Integração de Sistemas

Conforme visto no capítulo 2, a migração incremental do sistema legado exige que *gateways* sejam utilizados para que os dados, sistemas ou funcionalidades legadas que ainda vão

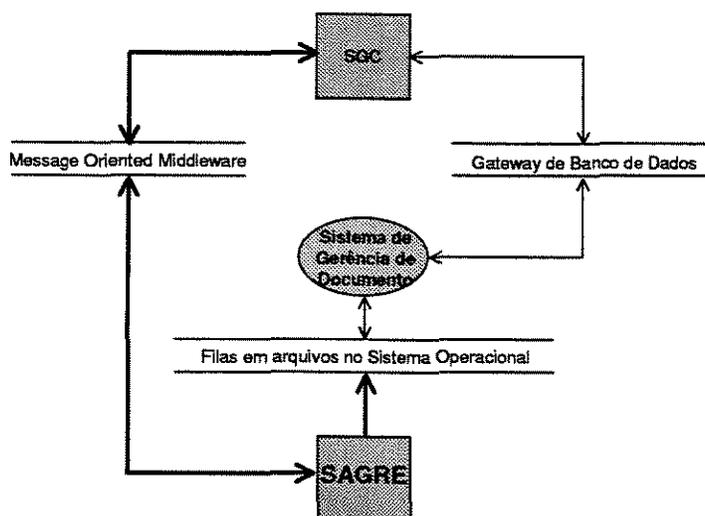


Figura 4.9: Arquitetura de Integração SAGRE e SGC

permanecer, possam ser acessíveis pelos novos sistemas. No caso da implantação do SAGRE na EOT não foi diferente, foi necessário o projeto de uma integração com o SGC para que os processos da empresa continuassem funcionando. A figura 4.9 apresenta a arquitetura de integração do SAGRE no ambiente da EOT.

Durante a fase de projeto da interface entre o SAGRE e o SGC foi colocado um novo requisito pela EOT: o módulo de gerência de documentos do SGC seria substituído por um novo sistema, o Sistema de Gerência de Documentos (SGD). Este sistema já estava implantado em outras empresas do grupo e iria diminuir praticamente todo o fluxo de papel que existia nos escritórios. Por se tratar de um sistema corporativo, todos os usuários da empresa poderiam acessar os documentos e a quantidade de impressão de cada OS ou BD seria em muito reduzido. O sistema também iria fazer todo o controle do fluxo dos documentos até o seu fechamento. Estas atividades já eram feitas pelo SGC mas de forma não trivial e qualquer mudança nos processos de negócio da empresa exigia

uma alteração no código do SGC. A proposta do SGD é que novos fluxos ou alterações nos atuais seriam apenas uma questão de configuração do sistema.

Desta forma o SAGRE seria integrado com o SGD e não mais com o SGC. O sistema de gerência de documentos trouxe uma arquitetura própria para integração de sistemas. Sua conexão com o *mainframe* se dá através de *gateways* de banco de dados. Esta solução, apesar de já estar operando em outras empresas do grupo se mostrou bastante ineficiente principalmente em relação ao desempenho. Alguns documentos chegavam a demorar uma hora para sair do *mainframe* e chegar no ambiente do SGD.

A integração do SAGRE com o SGD também foi feita utilizando-se as interfaces que o SGD já possuía. A proposta foi a implementação de filas em arquivos no sistema operacional. O SGD escreve no diretório *SGD-output* que é o mesmo diretório que o SAGRE faz a leitura dos documentos. Após o processamento o SAGRE escreve o resultado no diretório *SAGRE-output* que é o mesmo diretório que o SGD faz a leitura dos documentos para continuar o fluxo dos mesmos. Cada documento é escrito em um arquivo e cada tipo de documento tem uma interface distinta. O nome do arquivo tem sua regra de construção para que os programas de leitura possam fazer as verificações necessárias. Esta também não foi a melhor solução a ser adotada pois apresenta os seguintes problemas:

- não transparência de localização - dependência da configuração do sistema operacional para o compartilhamento de diretórios;
- desenvolvimento proprietário - implementação serviços de *pooling* para leitura dos arquivos e componentes para validação dos dados;
- desempenho - latência do esquema de *pooling*;
- falta de mecanismo de garantia de entrega da mensagem;
- falta de um mecanismo de recuperação de falhas;
- segurança - abertura do esquema de segurança do sistema operacional para permitir o mapeamento de diretórios.

A adoção de um *middleware* padrão de mercado, conforme havia sido sugerido, iria poupar esforços de implementação, além de fornecer todos os serviços necessários para uma solução mais robusta, com padrões abertos e prontos para que novos sistemas pudessem ser integrados à solução global da empresa. A solução atual somente está ativa porque os fluxos dos processos de negócio não exigem que as transações sejam síncronas, minimizando os problemas de desempenho e pela grande quantidade de técnicos que fazem a manutenção nos documentos com problemas ocasionados pelos erros na entrega/recebimento dos mesmos.

Durante o projeto da integração dos sistemas foi identificado um fluxo nos processos de negócio que necessitava uma interface síncrona entre o SAGRE e SGC. A atual solução via o sistema de gerência de documentos mostrou-se ineficiente em relação ao seu desempenho. Desta forma a implementação de uma integração direta tornou-se necessária. Após analisar as soluções de mercado, decidiu-se utilizar um *middleware* orientado a mensagens. Esta solução levou em consideração os seguintes fatores:

- robustez da solução - o *middleware* proposto é o principal produto para integrações entre ambiente Unix e *mainframe*;
- quantidade de fluxos a serem implementados - por ser necessário apenas um fluxo de integração não seria necessário utilizar um *middleware* com vários serviços;
- desempenho - apesar desta categoria de *middleware* não ser uma solução para integrações síncronas pois elas se baseiam em filas de transmissão e recepção, o seu forte desempenho associado a mecanismos de *time-out* garante uma solução adequada para a implementação deste fluxo;
- custo - esta foi a solução com melhor custo/benefício em termos de licença e manutenção
- existência de funções para implementar um mecanismo de recuperação de falhas;
- segurança - problemas de segurança seriam resolvidos pois a ferramenta se encarrega do transporte da mensagem, possibilitando uma certa transparência de localização dos serviços.

O sucesso desta implementação fez com que ela se tornasse padrão para novas soluções de integração na EOT e algumas integração problemáticas que existiam no ambiente da empresa começaram a ser migradas para esta nova arquitetura.

Arquitetura de Sistemas de Informação

O SAGRE é um produto para gerência da rede externa de telecomunicações. Ele utiliza-se de ferramentas e soluções georeferenciadas pois as atividades de gerência de rede assim o exigem. A solução SIG adotada faz uso do SGBD Oracle para armazenamento de todas as informações. A figura 4.10 apresenta a composição hierárquica dos componentes utilizados para a solução SAGRE.

A escolha do SIG deu-se principalmente pelas seguintes características:

- ferramenta que suporta o desenvolvimento de sistemas com modelos de dados complexos;

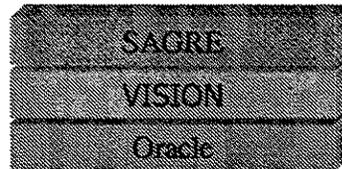


Figura 4.10: Composição hierárquica dos componentes do SAGRE

- regras de *renderização*⁶ configuráveis;
- gerência de esquemáticos;
- gerência de transações longas.

Além disto, a utilização de um SGBD comercial tornou a solução mais robusta em relação a:

- escalabilidade;
- banco de dados único;
- gerência de transações;
- desempenho para transações OLAP;
- *database gateways*;
- ferramentas de administração.

⁶regras de desenhos dos objetos como: tipo e tamanho de símbolos, cor e espessura de linhas, posicionamento de textos

Todos os módulos do sistema apresentam necessidades distintas em relação a operadores espaciais e abstração de dados. Porém em termos de projeto eles apresentam uma estrutura baseada em camadas. Eles foram projetados em três camadas (*three-tiers*) de abstração: camada apresentação, camada de negócios e camada de acesso a dados [Ma97].

A camada de acesso a dados é considerada o núcleo do sistema e sua modelagem foi feita utilizando conceitos de orientação a objetos e posteriormente sua tradução para o modelo relacional do SGBD comercial utilizado pelo SAGRE.

As camadas de negócios e de apresentação também foram modeladas utilizando ferramentas orientadas a objeto. Elas foram projetadas sob um metamodelo que é utilizado intensivamente por estas duas camadas. Isto possibilitou a construção do componente de Gerência de Atributos (GAT) dinâmico [OCMa95]. A figura 4.11 detalha a arquitetura do componente GAT.

Atualmente todos os módulos do SAGRE utilizam este componente com exceção do módulo de Operação devido a fatores históricos. A migração deste módulo para utilizar o componente GAT já está sendo projetada.

Outra característica importante da utilização de metamodelos é diminuição de linhas de código, rapidez no desenvolvimento de soluções e maior confiabilidade nos novos produtos por estar utilizando um componente já testado e robusto [RHH96, Riv97].

Uma possível desvantagem de utilização de sistemas baseados em metamodelos seria a sobrecarga nos SGBD diante do intenso acesso aos dados. Este problema foi solucionado utilizando um recurso da linguagem JAVA: a serialização. Este recurso permite a criação de um cache (um arquivo no sistema operacional) das informações que estão na memória. Desta forma ao invés da aplicação recuperar as metainformações do SGBD, ela irá fazê-lo a partir de um arquivo. Foi necessária apenas uma leitura das informações do banco de dados para gerar o cache e depois podem ser processadas inúmeras leituras do cache em arquivo.

Da mesma forma o SIG utiliza um cache de dados gráficos para armazenar os mapas em um formato já *renderizado* em arquivos do sistema operacional. O cache é composto por arquivos com informações que já passaram pelas regras de *renderização*. A área que corresponde ao banco de dados é toda quadriculada. O tamanho das quadrículas, ou mosaicos, é configurado pelo administrador do sistema. Cada mosaico corresponde a um arquivo no sistema operacional e pode conter informações de um ou vários objetos. A figura 4.12 apresenta a arquitetura da solução de cache de dados gráficos oferecida pelo SIG.

A solução proposta apresenta três níveis de hierarquia: cache principal, cache remoto e cache local. Esta classificação refere-se a localização geográfica dos caches na rede corporativa do cliente. O cache principal está sempre no mesmo *site* do SGBD. O cache remoto está localizado em um *site* remoto em relação ao cache principal e o cache local

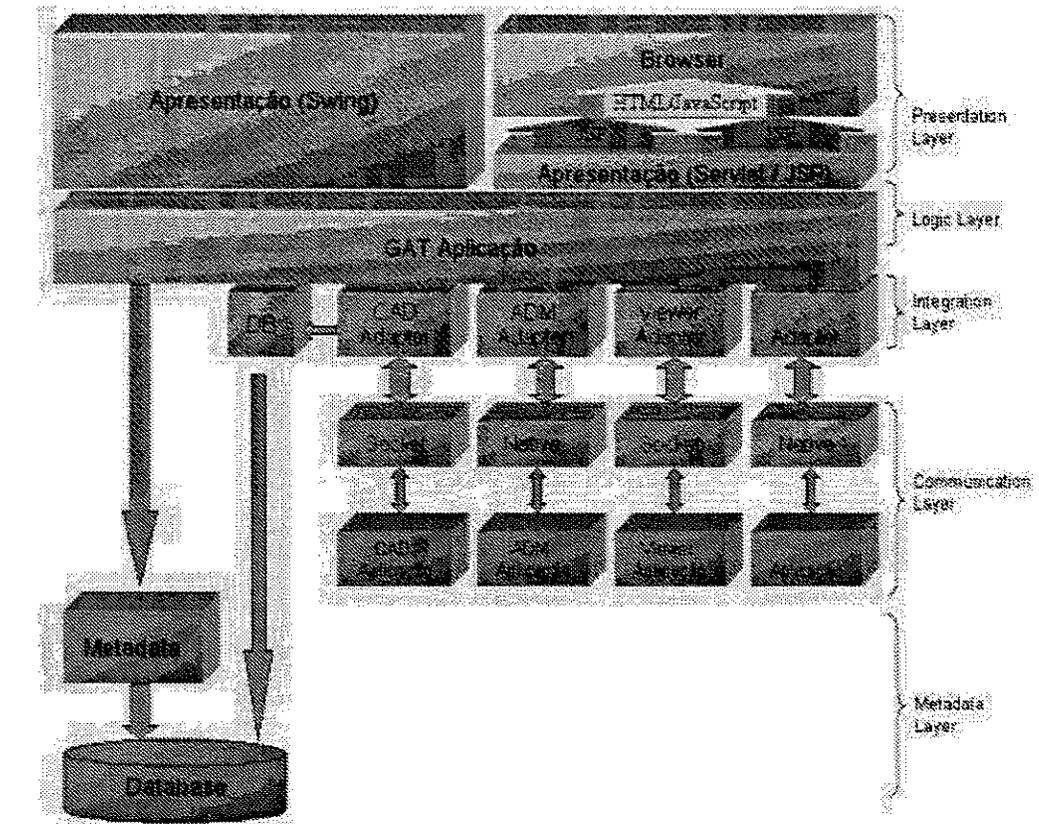


Figura 4.11: Arquitetura GAT

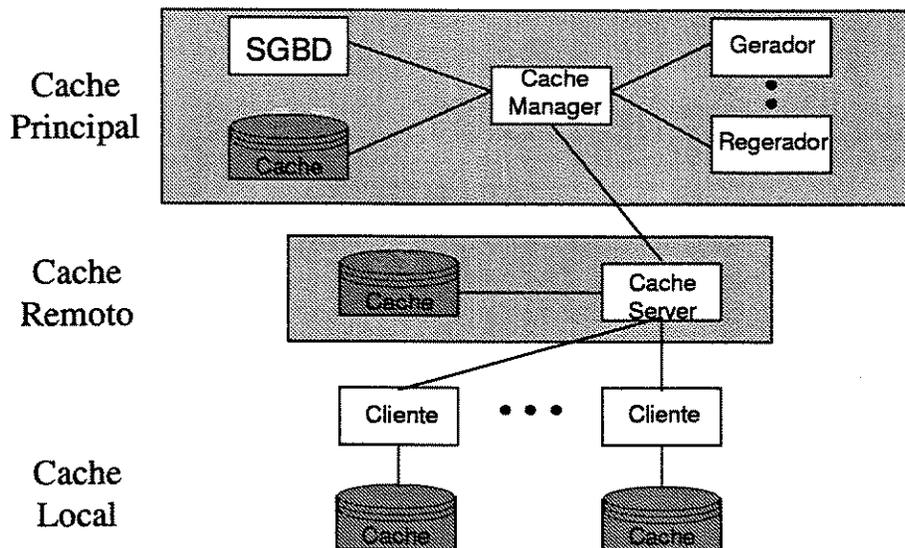


Figura 4.12: Arquitetura cache dados gráficos

deve estar no mesmo *site* dos módulos cliente do SAGRE. Dependendo da distribuição geográfica dos *sites* clientes e do *site* principal, da qualidade da rede e quantidade de banda disponível para o SAGRE o cache remoto pode não ser utilizado.

Existem dois serviços para fazer o gerenciamento do cache: o gerador e o regenerador. O gerador faz a geração completa de todos os mosaicos que compõem o cache enquanto que o regenerador faz a manutenção do cache, regenerando apenas os objetos gráficos que tiveram alguma alteração que afete sua regra de *renderização*.

Quando um cliente faz a carga dos mapas na tela da sua aplicação, ao invés da aplicação fazer o acesso ao SGBD, selecionar todos os dados da área que o usuário deseja trabalhar, aplicar as regras de *renderização* e somente depois mostrar os dados na tela, as aplicações fazem a leitura dos arquivos do cache e baseado no nome dos arquivos que possuem a semântica da área que eles representam, faz a leitura direta dos arquivos com dados *renderizados*.

Existe um outro serviço entre o cache principal e remoto: em todo acesso a um mosaico do cache remoto, é disparado um evento para checar a data do mosaico no cache principal e caso este seja mais recente, ele é copiado para o *site remoto*. A mesma regra é utilizada para a interação entre o *site* remoto e o *site* local.

Para que esta solução fique totalmente otimizada é necessário que a extensão que os mosaicos representam, seja de um tamanho adequado para que se tenha uma quantidade boa de dados em relação ao tipo de rede e localização geográfica dos arquivos.

Em relação à carga no SGBD a solução já está totalmente otimizada pois com uma leitura dos dados do SGBD e uma *renderização* é gerado o cache de dados gráficos que fica disponível para quantos acessos forem feitos sem a necessidade de uma nova *renderização*.

Por fim a arquitetura completa do SAGRE, com todos os seus componentes e produtos utilizados, é apresentada na figura 4.13.

Os módulos do SAGRE foram projetados para serem aplicações *desktop*. Porém o avanço da internet e dos produtos para desenvolvimento e disponibilização de dados via Web, direcionaram as novas implementações do SAGRE para esta nova plataforma. As antigas aplicações aos poucos vão sendo migradas de acordo com suas características e requisitos de abrangência e de desempenho. Já é possível fazer a visualização de mapas, atributos e raio de abrangência dos terminais públicos via Web. Também é possível visualizar imagens rasterizadas em vários formatos e todos os relatórios já foram migrados para serem visualizados via Web.

Na camada de serviços Web é utilizado o produto MapGuide da Autodesk para a visualização dos mapas. Ele faz a leitura de arquivos contendo mapas georeferenciados em diversos formatos e publica-os na Web. Os *browsers* devem ter a permissão para instalar *plug-in* ou componentes *ActiveX* que fazem a comunicação dos clientes com o servidor MapGuide. As tecnologias e servidores para ASP e JSP são utilizadas para a construção

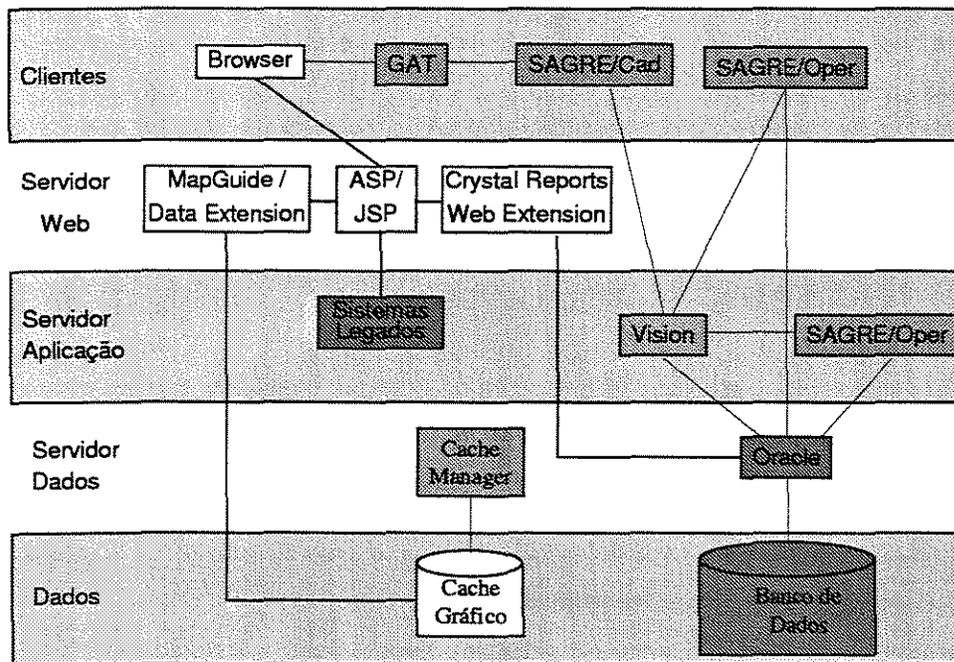


Figura 4.13: Arquitetura de componentes do SAGRE

das páginas de interação com o usuário para a navegação nas telas iniciais, para passagem de parâmetros de relatórios e para autenticação dos usuários. O componente Web do Crystal Reports é utilizado para a publicação de relatórios. Os servidores de página JSP foram utilizados para facilitar interface com os sistemas legados pois a tecnologia Java fornece várias formas de integração com outros sistemas seja via RMI, sockets, *middlewares* padrão CORBA ou J2EE.

A camada de aplicação fica reservada para o SIG Vision e Sistemas Legados. O SAGRE/Oper nesta camada é um reempacotamento do módulo de Operação sem a interface humano-computador. Nesta camada o SAGRE/Oper executa como um serviço de comunicação com o sistema de gerência de documentos. Foi desenvolvida uma nova camada de abstração que faz a monitoração de chegada de novos eventos (arquivos) vindos do SGD para processá-los e devolvê-los para o mesmo sistema. Este reempacotamento com uma nova camada de interface somente foi possível pelo projeto do sistema ter sido feito baseado em uma arquitetura de três camadas.

Já a camada de serviços de dados suporta o SGBD Oracle e os serviços para geração do cache gráfico. E por último a camada física de armazenamento dos dados.

4.5 Migração dos Sistemas Legados

Esta foi a fase mais longa e difícil de toda a migração pois neste momento o novo sistema foi detalhado, projetado e construído. Nesta fase todas as etapas do ciclo de vida de desenvolvimento de um sistema estão presentes. Está fora do escopo deste trabalho detalhar todo o desenvolvimento do módulo de Operação do SAGRE. A seguir serão apresentadas as principais atividades e alguns exemplos da construção e implantação do módulo SAGRE/Oper seguindo os passos propostos pela metodologia.

4.5.1 Detalhar o Sistema Legado

Nesta etapa foi realizado um maior detalhamento do sistema legado para ser possível o projeto do novo sistema de informação que iria substituí-lo.

Classificar o Sistema Legado

O sistema de gerência de facilidades foi classificado como um sistema semi-estruturado pois em determinadas partes a camada de apresentação era bem isolada das camadas de aplicação e dados e em outras a camada de dados estava desacoplada das camadas de apresentação e aplicação. Ele foi todo escrito em linguagem NATURAL e utilizava algumas funções em COBOL do sistema de gerência de clientes.

O SGF pode ser dividido em quatro grandes partes:

- Cadastro e manutenção de facilidades de rede;
- Gerência de facilidades;
- Emissão de projetos;
- Relatórios.

O cadastro e manutenção de endereços era feito pelo SGC e disponibilizado para o SGF através de visões no SGBD. O acoplamento entre os dois sistemas era baixo pois o SGF fazia uma cópia das informações de endereço para seus próprios campos, replicando os dados em relação ao SGC. Porém a coesão era baixa uma vez que não existia nenhum mecanismo de sincronização de informações de endereçamento do SGC em relação ao SGF no momento de alterações por parte do SGC.

Identificar as Regras de Negócio implementadas pelo Sistema Legado

Existiam dezenas ou centenas de regras de negócio na semântica do código do SGF. Nenhuma delas estava explicitamente documentada ou encapsulada dentro do código. Durante a construção do SAGRE/Oper essas regras foram sendo colhidas através de reuniões com os líderes de desenvolvimento e mais recentemente elas têm sido verificadas com os usuários finais do SGF. Algumas exemplos de regras são:

- não pode ser designado um circuito se as facilidades utilizadas por ele estiverem em projeto;
- o mesmo par não pode estar sendo ocupado por dois circuitos ao mesmo tempo;
- um par somente pode conter um cliente de telecomunicações, a menos que o par tenha um equipamento de multiplexação associado;
- um cliente de telecomunicações deve ser atendido pela facilidade de rede que esteja livre e mais próxima a seu endereço;
- durante a fase de construção ou reparo da rede de telecomunicações, os clientes atuais não podem ficar com seus serviços indisponíveis.

Identificar e Classificar os Dados do Sistema Legado

Uma das atividades mais difíceis de toda a migração foi a coleta e entendimento dos dados do sistema legado. O gerenciador de banco de dado do SGF era o ADABAS dificultando ainda mais a compreensão das estruturas de dados por ele não possuir ferramentas

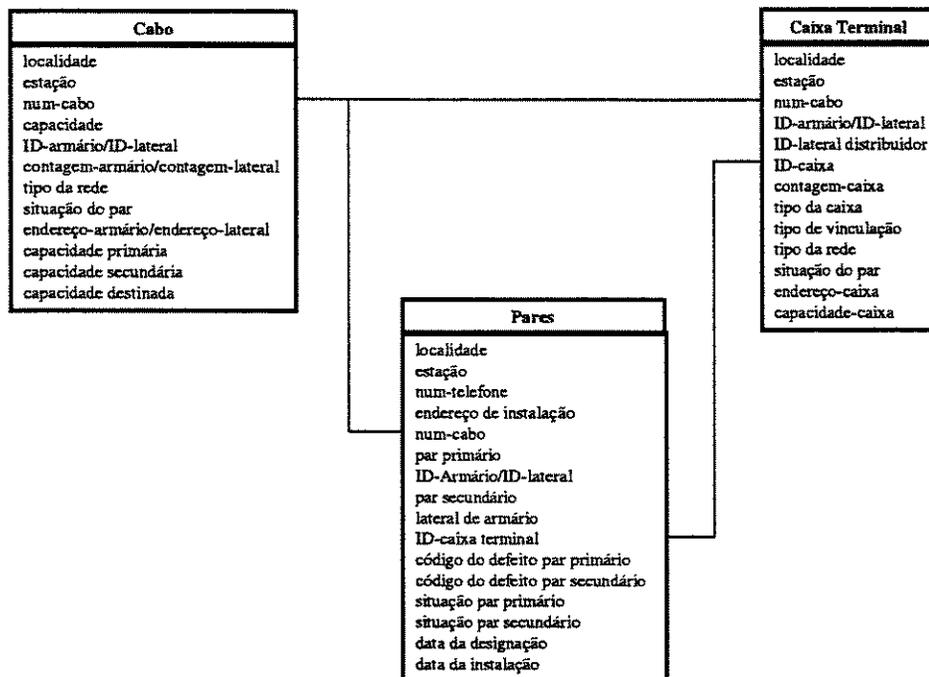


Figura 4.14: Modelo de Dados do SGF

amigáveis, gráficas para mostrar as relações entre os dados. Este trabalho seria praticamente impossível se uma parte da equipe de administração do banco de dados do SGF não tivesse cooperado com o trabalho.

A figura 4.14 apresenta um esquema de relacionamento entre os principais objetos do banco de dados do SGF. Ele já foi traduzido para o modelo de dados relacional que é o modelo de dados do SAGRE, portanto já está no formato canônico.

Depois da fase de exportação do modelo de dados local para o modelo de dados canônico foi feita uma análise de cada atributo e a sua relevância ou não para o futuro sistema. Para fazer esta análise foram criadas matrizes de classificação dos dados. A tabela 4.2 apresenta a matriz de classificação para a tabela de Caixa Terminal do sistema legado.

Pela análise da tabela de Caixa Terminal ficou claro que nem todos os dados do sistema legado deveriam ser convertidos para o SAGRE. Neste caso os atributos tipo de vinculação

Caixa Terminal	Sistema	Configuração	Transação	Apoio Decisão	Descartável
localidade	X		X		
estação	X		X		
num-cabo	X		X		
ID-armário/ID-lateral	X		X		
ID-lateral distribuidor	X		X		
ID-caixa	X		X		
contagem-caixa	X		X		
tipo da caixa		X	X		
tipo de vinculação					X
tipo da rede					X
situação do par	X		X		
endereço-caixa		X	X		
capacidade-caixa		X	X		

Tabela 4.2: Matriz de classificação dos dados de Caixa Terminal

e tipo da rede poderiam ser obtidos através de outros relacionamentos ou da forma com que a rede externa de telecomunicações foi modelada no SAGRE. Também o atributo de tipo da caixa e capacidade da caixa deveria pertencer a outra tabela, mais completa, que tenha todas as informações padrões do objeto Caixa Terminal.

No caso do atributo endereço-caixa existe uma particularidade pois como o SAGRE é um produto baseado em um sistema de informações geográficas, este relacionamento para o SAGRE é facilmente obtido através de uma operação espacial.

No caso do SGF todos os dados foram considerados como dados transacionais, não tendo nenhum dado de apoio à decisão.

Este tipo de análise foi realizada para todas as tabelas do SGF.

4.5.2 Projetar e Construir o Sistema de Informação Destino

Após a análise detalhada do sistema legado foram iniciadas as atividades de projeto do novo sistema de gerência de facilidades. Os principais requisitos que deveriam ser preenchidos com o novo sistema eram:

- integração entre os ambientes de engenharia e operação;
- integração com o sistema de gerência de documentos;
- escalabilidade do sistema;

- interface amigável para os usuários.
- agilidade no gerenciamento de novos serviços de telecomunicações;

Integração entre os Ambientes de Engenharia e Operação

A integração entre os ambientes de engenharia e operação eram necessárias porque na situação em que a EOT se encontrava, todos os projetos eram feitos em papel ou em ferramentas CAD sem nenhum critério nem padrão. Não existia um repositório único para os arquivos resultantes dos projetos feitos nas ferramentas CAD. Pelo fato dos projetos serem terceirizados também não existia uma padronização em relação às ferramentas CAD utilizadas. No final estes projetos feitos em ferramentas CAD eram impressos e guardados nas mapotecas em papel.

Nos projetos de *expansão* da rede, onde não é necessário alterar a rede existente, era necessário fazer a atualização do cadastro de facilidades da rede no sistema SGF. Esta atualização era feita por técnicos que deveriam *ler* as plantas com o projetos executados em campo e cadastrar os dados no SGF. Como o sistema de gerência de facilidades não é um sistema georeferenciado, o processo de designação das melhores facilidades para cada usuário baseado na distância entre o usuário e os equipamentos de atendimento ao cliente era feito de maneira aproximada, isto é, no momento do cadastramento dos equipamentos o técnico deveria identificar o endereço de instalação do mesmo e definir a faixa de endereço coberta pelo equipamento. Este processo era repleto de inconsistências por vários fatores:

- mapas com informações incorretas, principalmente em relação ao Mapeamento Urbano Básico (MUB);
- mapas não georeferenciados corretamente induziam a interpretações erradas em relação à real posição dos equipamentos e conseqüentemente erros na definição da área de cobertura dos equipamentos;
- erros de digitação;
- erros de interpretação dos mapas;
- alterações no nome e numeração dos logradouros dificilmente eram refletidas no cadastro.

Os projetos de *alívio* de rede são bem mais complexos pois além dos problemas técnicos envolvidos no projeto de rede, deve-se levar em consideração os usuários de telecomunicações que estão ativos. Existia um relatório específico gerado pelo SGF, o Relatório de Emissão de Projetos (REP) que detalhava os procedimentos que devem ser executados

para que os usuários não fiquem com o serviço indisponível por um período maior que alguns minutos. Quando os usuários são grande empresas este tempo de indisponibilidade não deve existir, para isto são tomadas outras decisões de execução de projeto além das utilizadas para usuários residenciais.

Com a utilização dos módulos de cadastro, projeto e operação do SAGRE estes problemas foram resolvidos pois a proposta do SAGRE é de ser um sistema integrador de todas estas áreas de empresa e das fases do ciclo de projeto e construção de rede. O processo de construção dos mapas georeferenciados para inicializar o banco de dados do SAGRE foi um processo longo e com altos custos porém a empresa já está colhendo os resultados deste investimento. As áreas em que o SAGRE/Oper está em produção não é necessário fazer a *tradução* do projeto em papel como era feito no SGF. O SAGRE com seus componentes espaciais é capaz de fazer a designação por geografia das facilidades de rede que atendem os pedidos dos usuários de telecomunicações.

Integração com o Sistema de Gerência de Documentos

O outro requisito importante era a integração do novo sistema de gerência de facilidades com o sistema de gerência de documentos. Este sistema estava substituindo a parte de *workflow* do sistema de gerência de clientes e seria o integrador entre todos os sistemas de empresa na área de Sistemas de Suporte a Operação (SSO).

Conforme descrito na seção anterior a solução de integração foi definida pelas equipes do SGD e o SAGRE/Oper precisou se adaptar a esta forma de integração. Nesta fase foram detalhados todos os fluxos de integração, as assinaturas dos eventos de integração, tratamentos de exceção, os eventos e procedimentos de recuperação de falhas, os requisitos de desempenho e configuração do ambiente operacional para que os sistemas pudessem interoperar.

A figura 4.15 apresenta um diagrama de sequência de um dos eventos de integração.

Para que este requisito fosse possível o módulo de operação foi projetado utilizando a arquitetura em três camadas conforme descrito na seção anterior.

Escalabilidade do Sistema

O sistema legado de gerência de facilidade é executado no ambiente *mainframe* sua capacidade de processamento é da ordem de milhões de instruções por segundo. Este ambiente é muito estável, rápido e robusto porém da forma como o sistema foi projetado não favoreceu o aspecto de escalabilidade tornando os custos para aumento da capacidade de processamento e manutenção muito elevados. O novo sistema deveria apresentar as mesmas características de volume de processamento e robustez que o sistema legado.

Para atingir estes objetivos foi elaborada a arquitetura de rede detalhada na figura

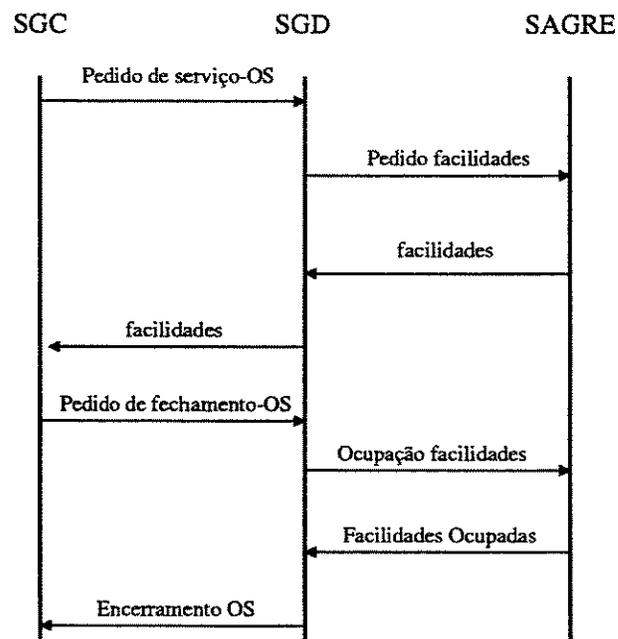


Figura 4.15: Diagrama de Sequência do fluxo de Abertura e Fechamento de OS

4.8. Foram feitos experimentos de troca de pacotes na rede em relação ao tamanho e volume dos pacotes para definir a quantidade de banda necessária e latência máxima da rede. Também foram projetados servidores de dados em um ambiente em *cluster* de alta disponibilidade. O tamanho dos servidores de dados, aplicações, Web e dos clientes foram especificados de acordo com as necessidades de cada área usuária.

O SAGRE adotou uma política de divisão dos bancos de dados por região para resolver uma parte dos problemas de escalabilidade do sistema. Isto não impediu o correto funcionamento do sistema uma vez que a gerência das facilidades de rede na grande maioria dos casos está restrita à área compreendida por uma cidade ou localidade. Desta forma cada servidor de dados e aplicações é responsável por uma área de atuação. Isto também não impactou a área de engenharia da empresa, no caso o módulo de Cadastro e Projeto do SAGRE, pois os projetos de rede de acesso⁷ abrangem no máximo a área de uma cidade.

Foi necessário a criação de alguns roteadores pois como o sistema de gerência de clientes também apresenta uma base de dados única para toda a empresa os eventos de integração entre os sistemas sempre utilizam a mesma porta de entrada (diretório no sistema operacional). Os roteadores construídos fazem o encaminhamento dos pedidos do SGC para os servidores de dados apropriados do SAGRE.

Interface Amigável para os Usuários

O requisito de interface humano-computador era muito importante nesta migração pois a mudança de um sistema baseado em terminais, com interface caracter para um ambiente gráfico e amigável era muito importante [Mer95]. Toda a interface foi implementada em uma ferramenta de geração de interface que é capaz de gerar código para a maioria das plataformas Unix e para o ambiente Windows. A figura 4.16 apresenta uma tela do sistema.

Agilidade no Gerenciamento de novos Serviços de Telecomunicações

A necessidade de um sistema configurável é resultante do aumento da oferta de novos serviços de telecomunicações e conseqüentemente necessidade de gerenciá-los. O novo sistema deveria ser capaz de não apenas gerenciar os atuais serviços suportados pelo SGC mas também gerenciar novos serviços como, por exemplo, linhas digitais assíncronas, isto é, nas mesmas facilidades de rede existentes para um usuário residencial ou comercial de pequeno porte, oferecer serviços de acesso a internet com melhores taxas de transmissão.

Para atender todos os requisitos do novo sistema o módulo de operação do SAGRE foi dividido em quatro subsistemas:

⁷rede para atendimento dos usuários de telecomunicações

ATRIBUICAO DE DEFETTO

Arquivo Edição

Cabo: [] [] [] [] H I

Falta de Peças: [] * [] I

Estado: [] [] [] [] H I

Armadão: [] [] [] [] H I

Tipo Defeito: [] * [] H I

Data Detecção: [] [] [] [] I

Observações: [] I

Tipo Def. : [] * [] H I

Nº. Def. : [] [] [] [] I

Data Intação: [] [] [] [] I

OK CANCELAR

Figura 4.16: Tela do SAGRE/Oper

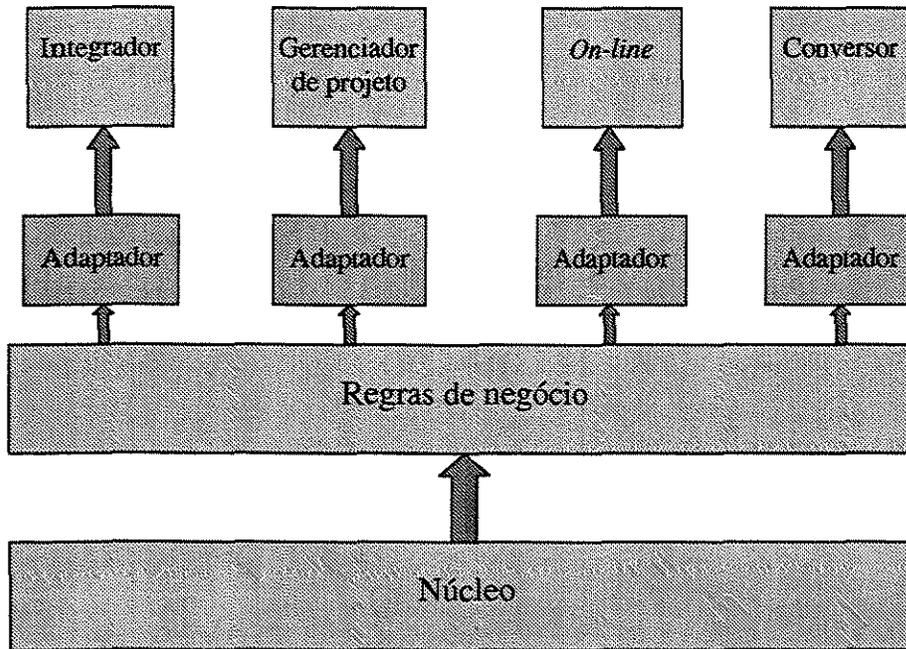


Figura 4.17: Arquitetura dos subsistemas do módulo de operação

- *on-line* - esta é a parte de interação direta com o usuário para as atividades de gerência de facilidades que não foram processadas automaticamente pelos eventos automáticos de integração com o sistema de gerência de clientes;
- integrador - esta é a parte de integração do SAGRE com o SGD para os eventos de integração automática com o SGC;
- conversor - esta é a parte do sistema que faz a leitura dos dados do sistema legado e faz a carga de dados inicial do SAGRE/Oper;
- gerenciador de projeto - este é o subsistema que faz a integração do módulo de cadastro e projeto do SAGRE com o módulo de operação.

A figura 4.17 detalha a estrutura do módulo de operação do SAGRE, seus subsistemas e integrações.

Para atender as necessidades de um sistema aberto e flexível para novos elementos e serviços de rede, foi projetado um modelo de dados simples para que fosse possível

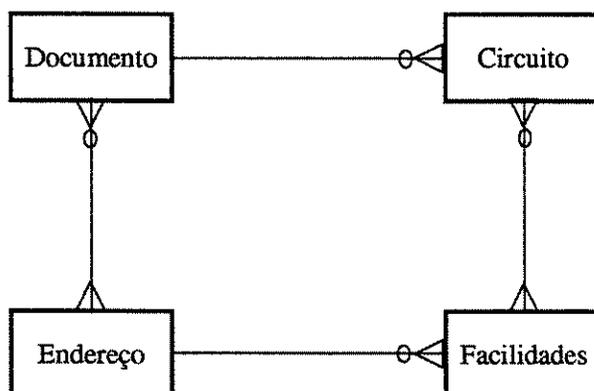


Figura 4.18: Modelo de dados - SAGRE/Oper

contemplar qualquer nova situação. A figura 4.18 apresenta um diagrama de entidade e relacionamento do modelo de dados do SAGRE/Oper.

Os documentos estão associados a nenhum, um ou mais circuitos dependendo do seu tipo e estado do mesmo. Os circuitos estão associados a um documento. Circuitos são compostos por várias facilidades de rede e as facilidades estão associadas a nenhum, um ou vários circuitos. Os endereços estão associados a nenhum, um ou várias facilidades de rede e/ou documentos. Uma facilidade de rede está associada a um endereço e um documento está associado a um ou vários endereços.

O esquema do banco de dados do módulo de operação do SAGRE, no modelo relacional, é apresentado na figura 4.19.

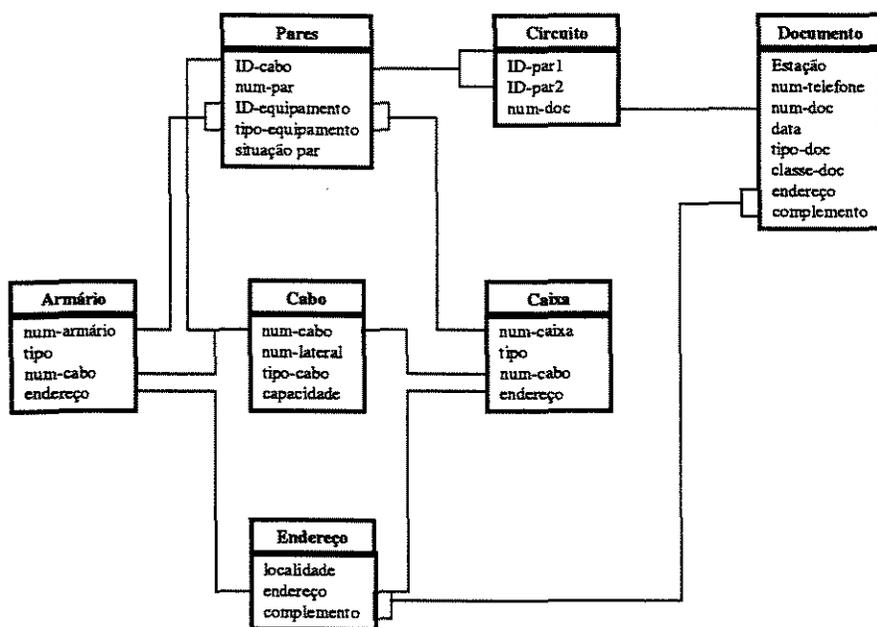


Figura 4.19: Modelo de Dados SAGRE/Oper

4.5.3 Projetar e Construir Ferramentas para Conversão dos Dados do Sistema Legado para o Sistema Destino

A etapa de conversão de dados em qualquer projeto é uma das etapas mais complexas e longas. Para projetos baseados em sistemas georeferenciados esta é a parte mais crítica de toda a implantação do sistema. Sem mapas de qualidade e com informações mínimas, o sistema pode não dar o retorno esperado pela empresa e usuários. Porém a aquisição de mapas com boa acurácia e em escalas apropriadas para sistemas de suporte a operação são difíceis e demorados de serem preparados. A correta especificação dos elementos que devem compor os mapas foi fundamental nesta fase.

O SAGRE apresentou soluções inovadoras na área de conversão de dados georeferenciados. O formato de conversão SAGRE, ou formato DAT como é referenciado pelas empreiteiras de rede, e a classificação dos tipos de MUB para as diversas aplicações garantiram uma padronização dos mapas, formatos de conversão e formato de dados no mercado brasileiro de conversão de dados.

Tradicionalmente as empresas de conversão de dados do mundo todo realizam a conversão baseadas no formato de dados do SIG utilizado pelo cliente. Isto requer um alto grau de especialização das empresas de conversão que necessitam contratar técnicos especializados para a elaboração dos mapas pois eles precisam entender do formato dos diversos SIG comerciais, construção de ferramentas para geração dos dados nos diversos formatos dos SIG e necessidade de conhecer detalhadamente o modelo de dados de cada aplicação para as quais eles estão fazendo a conversão. Todas estas atividades são demoradas e caras, muito mais para um mercado de conversão de dados que não possuía empresas de porte e com conhecimento do assunto, como o brasileiro.

O SAGRE apresentou um novo paradigma para o mercado. Foi proposto um modelo de dados canônico, independente do formato do SIG utilizado e com a representação em alto nível do modelo de dados. Esta abordagem garante um baixo acoplamento do modelo de conversão em relação ao modelo de dados dos sistema, garantido uma liberdade para evolução do sistema sem prejudicar contratos de conversão que possam estar em andamento. Esta abordagem também colocou todas as empresas de conversão no mesmo patamar, não privilegiando grandes empresas que poderiam rapidamente criar os formatos específicos do SIG, impedindo as pequenas empresas de entrarem no processo e consequentemente diminuindo a concorrência.

A seguir é apresentada uma parte da especificação do formato SAGRE e um exemplo de do objeto Cabo (031) do módulo de cadastro:

031 A xpa, ypa

onde

xpa: coordenada x de identificação de um dos ponto de acesso do cabo

ypa: coordenada y de identificação de um dos ponto de acesso do cabo
 ...
 031 E num_emendas, ano_instalação, situação, tipo_cabo, bitola, capacidade
 onde
 ...
 031 A 606186.224, 7801658.163
 031 A 606123.980, 7801806.122
 031 A 606380.102, 7801938.165
 ...
 031 B 606477.042, 7801099.490
 031 C 606663.265, 7801989.796
 ...
 031 D 606380.102, 7801938.490
 031 E 0, ,5, "CA", 50, 10
 031 F 8, "B1Q", "VIV", , "03", , 1, 30, 2, 1
 031 F , , , , , 1, 50, 3, 2
 031 G 1, 0, 606997.449, 7801984.694, 45

Outra importante padronização do projeto SAGRE foi a classificação dos tipos de MUB. Cada módulo do sistema necessita de um tipo e precisão de dados georeferenciados. Eles podem trabalhar com dados de MUB mais ou menos detalhados sem perda da qualidade da informação. A seguir são apresentado os tipos de MUB e a suas composições:

- MUB1 - limites de cidades e limites de mancha urbana;
- MUB2 - trecho de logradouro, bairros, nomes de logradouros;
- MUB3 - quadras, intervalo de numeração por trecho de logradouro, CEP por trecho de logradouro;
- MUB4 - divisa de lotes e endereços;
- MUB5 - classificação de mercado por endereço.

Além destes elementos, considerados essenciais para cada tipo de aplicação ainda existem os elementos opcionais como: hidrografia, limite de estado, meio-fio, ferrovia, rodovia, etc.

A migração dos dados do SGF para o SAGRE/Oper foi feita utilizando o formato SAGRE. A equipe do SGF desenvolveu uma ferramenta de exportação de dados baseada

no formato SAGRE para todos os tipos de documentos existentes no SGF. Desta maneira a migração dos dados do sistema legado para o sistema novo sistema pôde ser feita de maneira rápida e segura. De acordo com a arquitetura dos componentes do módulo de operação (fig. 4.17) todas as regras de aplicação e de banco de dados da versão do produto *on-line* estão presentes no produto de conversão.

Outra importante ferramenta construída foram relatórios a partir dos arquivos de *log* gerados pela ferramenta de conversão. Os arquivos de *log* foram projetados para imprimirem palavras-chaves para que posteriormente a ferramenta de relatório pudesse extrair, agrupar e sumarizar as informações relevantes para os usuários.

4.5.4 Definir o Plano de Implantação do Ambiente Final e Migração do Sistema Legado

Esta foi a fase mais operacional de toda a migração porém não menos importante. Todo o trabalho feito até este passo pederia ser literalmente invalidado caso esta fase não fosse bem planejada e executada. Os usuários criariam uma impedância na utilização do novo sistema caso alguma coisa saísse errado.

Durante esta fase foram disponibilizadas as seguintes equipes:

- equipe de implantação - responsável por planejar, organizar, viabilizar e acompanhar a entrada do SAGRE na EOT. A equipe deveria atuar de forma matricial em relação às equipes de desenvolvimento, ambiente operacional, suporte e treinamento
- equipe de ambiente operacional - equipe responsável por instalar e configurar os *software* básicos como SIG, SGBD e o SAGRE. Também foram os responsáveis por configurar o ambiente como um todo para garantir o desempenho máximo dos sistemas
- equipe de conversão de dados - esta equipe era dividida em duas frentes: equipe conversão de mapas e equipe de conversão dos dados do sistema legado para o SAGRE
- equipe de suporte - equipe responsável por auxiliar os usuários nas primeiras semanas de uso do sistema *in-loco* e também através de suporte remoto. Caso algum erro do sistema fosse identificado, a equipe de desenvolvimento era imediatamente acionada para a pronta correção

A estratégia de implantação foi bem elaborada. A equipe de implantação ficou encarregada de criar um documento chamado Plano de Implantação. Este documento continha todas as atividades necessárias para a correta implantação do sistema. Nele foram definidas as atividades, responsáveis, data de execução e previsão de duração das mesmas.

Também foram identificados os riscos e os procedimentos para contorná-los. A equipe de implantação ficou encarregada de acompanhar o cronograma de execução das tarefas e identificar desvios. Caso eles ocorressem a equipe de implantação deveria tomar ações para minimizar o impacto dos riscos no andamento do processo como um todo.

Nesta fase foi realizado o treinamento dos usuários pois o correto entendimento e uso do sistema diminuiria as barreiras de utilização do mesmo e diminuiria a quantidade de chamadas para a equipe de suporte remota, como de fato ocorreu.

Durante este processo duas estratégias se mostraram as mais importantes: implantação dos módulos de cadastro e projeto nos escritórios de engenharia e posteriormente a implantação do módulo de operação nos escritórios de operação. A outra estratégia importante foi a realização de um projeto piloto de implantação do módulo de operação.

A implantação dos módulos de forma incremental, iniciando com o módulo de cadastro e projeto, foi importante para provar alguns conceitos. Os usuários de engenharia não possuíam um sistema como o SAGRE, apenas utilizavam ferramentas CAD de forma *ad-hoc*. A vinda do sistema trouxe uma grande organização e padronização aos trabalhos, mas por outro lado impediu os usuários de fazerem determinadas atividade não padronizadas, como algumas plotagens, que estavam habituados devido a maior flexibilidade que tinham com as ferramentas CAD. Também foram identificadas necessidade de mudanças em algumas funcionalidades que não se mostraram produtivas no trabalho diário. Outras tiveram que ser implementadas como métodos de *rippling*⁸.

Para o projeto piloto do módulo de operação foi escolhida uma área telefônica com um volume médio e homogêneo de transações diárias. Também foram criados mecanismos de colocar o SGF em paralelo à execução do SAGRE/Oper para que, caso ocorresse qualquer problema, ele pudesse assumir a operação da rede. Isto possibilitou a estabilização o sistema e implementação de funcionalidades não detectadas durante a fase de especificação dos sistemas.

4.6 Resumo do Capítulo

Neste capítulo foi realizada uma aplicação da metodologia de migração de sistemas legados em ambiente de projeto e operação de uma Empresa Operadora de Telecomunicações. A empresa possuía quatro grandes sistemas legados nestas áreas: Sistema de Gerência de Clientes, Sistema de Gerência de Facilidades, Sistema de Demanda Telefônica e Sistema de Materiais e Mão de Obra. Os processos de negócio das áreas de engenharia e operação da empresa foram mapeados e posteriormente foi feita a priorização da migração dos sistemas legados.

⁸classe de algoritmos para encaminhamento em rede e propagação automática de informações comuns aos objetos ao longo do percurso

O SAGRE foi escolhido para automatizar os processos de cadastro e projeto de rede que não tinham uma ferramenta única ou muitas vezes estas atividades eram feitas utilizando-se apenas plantas em papel. Esta etapa não foi considerada uma migração de sistemas legados pois as ferramentas utilizadas não se enquadravam na categorização de sistemas legados que foi utilizada nesta dissertação. O módulo de operação do SAGRE foi escolhido para substituir o sistema de gerência de facilidades da empresa.

A partir da escolha realizada foi detalhada a arquitetura e dados do sistema legado, elaborada a arquitetura para o novo sistema que iria substituir o SGF, o SAGRE/Oper, e que pudesse ser utilizada para todos os outros sistemas da empresa que iriam substituir os demais sistemas legados. A arquitetura de integração proposta foi de certa forma limitada por requisitos de novos sistemas que estavam sendo implantados porém em alguns fluxos de integração foi possível estabelecer uma arquitetura de integração mais flexível. A arquitetura de rede e de sistemas de informação propostas foram plenamente aceitas e implantadas gerando resultados satisfatórios para a empresa.

Por último foi realizado o desenvolvimento e implantação do novo sistema seguindo os passos da última fase da metodologia proposta. Existiam grandes desafios nesta fase principalmente em relação ao levantamento dos dados do sistema legado, o projeto do novo sistema para atender todas as expectativas da empresa principalmente em relação aos requisitos de desempenho e estabilidade uma vez que o sistema legado se encaixava no conceito de sistema de missão crítica. Também a etapa de conversão dos dados do sistema legado para o novo sistema necessitou de muito planejamento, ferramentas específicas e uma grande equipe para realizar a limpeza e conferência dos dados. Finalmente os módulos de Cadastro, Projeto e depois de Operação foram implantados, também com muito planejamento, de forma incremental.

Capítulo 5

Conclusões

O principal objetivo desta dissertação foi definir uma metodologia para migração de sistemas legados em ambientes de projeto e operação. Foram definidos os objetivos, abordagens e possíveis restrições a serem seguidas em cada etapa da migração.

A metodologia apresentada está baseada no conceito de uma migração incremental. Vários trabalhos apontavam para esta abordagem [BS95], [GB95], [BNS97], e após a análise das opções, esta nos pareceu ser a melhor para a migração de sistemas legados complexos das grandes organizações. A metodologia fornece os subsídios para migração de qualquer tipo de sistema legado pela quantidade e variedade de tópicos abordados, indo desde a fase de mapeamento dos processos de negócio da organização, passando pela definição da arquitetura de sistemas de informação, projeto e implementação dos novos sistemas, até a fase de implantação dos sistemas no ambiente destino. Para sistemas legados com um grau de complexidade menor deve-se fazer um avaliação dos passos da metodologia e propor uma redução na quantidade ou escopo de cada passo, de forma a agilizar o processo de migração mas sem perder o objetivo da metodologia que é fazer a migração tendo o contexto geral dos processos da empresa e sua arquitetura de sistemas de informação.

A validação da metodologia foi realizada durante o planejamento e execução de uma migração em um caso real. Para a aplicação em um caso real foi escolhida uma Empresa Operadora de Telecomunicações que possui alguns sistemas legados de grande complexidade e que estavam em fase de implantação do SAGRE, um sistema de informações geográficas para gerência da rede externa de telecomunicações. Esta aplicação prática da metodologia auxiliou na tomada de decisões sobre os passos da metodologia como a inclusão da fase de implantação do novo sistema que iria substituir o sistema legado no ambiente destino.

As principais contribuições apresentadas por esta dissertação foram:

- análise das várias metodologias de migração de sistemas legados;

- proposta de uma metodologia que cobre todos os passos necessários para uma migração real;
- organização em forma de passos bem definidos das fases de mapeamento de processos de negócio e definição da arquitetura de sistemas de informação na metodologia proposta por Brodie et al. [BS95];
- aplicação da metodologia em um caso real de uma empresa operadora de telecomunicações utilizando um produto comercial como o SAGRE.

As propostas de migração de sistemas legados apresentadas na literatura abordam vários aspectos da migração porém nenhuma tem a abrangência apresentada neste trabalho. Apesar da aplicação prática da metodologia estar direcionada para ambientes de projeto e operação, ela não invalida a aplicação da metodologia em outros tipos de ambientes.

A existência de um sistema real, como o SAGRE, e uma aplicação prática da metodologia também em um ambiente real, possibilitou a comprovação das propostas apresentadas na metodologia.

As extensões propostas a este trabalho são:

- realizar uma aplicação prática da metodologia em sistemas legados de menor porte e fazer os ajustes nos passos da metodologia;
- realizar uma aplicação prática da metodologia onde exista uma plataforma de integração de sistemas única para toda a empresa;
- especializar a metodologia para Sistemas de Informação Geográficas;
- aplicar outras técnicas de mapeamento de processos;
- utilizar ferramentas para projeto da arquitetura de sistemas;
- criar ferramentas para automatizar o mapeamento dos dados.

Novas aplicações da metodologia vão garantir que as etapas propostas são realmente abrangentes e úteis para qualquer situação que possa ocorrer durante a migração de sistemas legados. Possíveis situações não cobertas poderão ser acrescentadas na metodologia tornando-a ainda mais abrangente. A aplicação em empresas onde exista uma plataforma única de integração de sistemas irá mostrar a flexibilidade da metodologia e provar os conceitos apresentados no capítulo 3 desta dissertação. Neste tipo de aplicação também poderá ser utilizada ferramentas para projeto da arquitetura de sistemas de forma a agilizar esta etapa da migração.

A fase de mapeamento de dados é uma das mais importantes de metodologia e requer um grau de automação maior. As ferramentas de modelagem de dados existentes hoje são muito boas para fazer engenharia reversa de diversos SGBD comerciais porém não dão suporte para um estudo mais detalhado da semântica dos dados.

Por fim a extensão da metodologia para Sistemas de Informação Geográficas seria interessante pois estes sistemas possuem características especiais no tratamento dos dados como relacionamentos espaciais, tipos de dados não convencionais, além de restrições de desempenho específicas.

Referências Bibliográficas

- [AAR94] P. Aiken, Muntz A., and R. Richards. Dod legacy systems - reverse engineering data requirements. *Communications of the ACM*, 37(5), Maio 1994.
- [Agu95] C. D. Aguiar. Integração de sistemas de banco de dados heterogêneos em aplicações de planejamento urbano. Master's thesis, UNICAMP, 1995.
- [AOS⁺00] A. Arulanthu, C. O'Ryan, D. Schmidt, M. Kircher, and J. Parsons. The design and performance of a scalable orb architecture for corba asynchronous messaging. In *Middleware 2000 Conference, ACM/IFIP*, April 2000.
- [Ber96] P. A. Bernstein. Interoperability. *Communications of the ACM*, 39(2):86–98, Março 1996.
- [BF96] G. Bracchi and C. Francalanci. A framework for the alignment of it architectures with information processing requirements of organizations. In *Americas Conference on Information Systems*, Agosto 1996.
- [BNS97] J. K. Bergey, L. M. Northrup, and D. B. Smith. Enterprise framework for the disciplined evolution of legacy systems - cmu/sei-97-tr-007. Technical report, Carnegie Mellon University, 1997.
- [BS95] M. Brodie and M. Stonebraker. *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach*. Morgan Kaufmann Publisher, Inc., 1995.
- [Car98] A. P. Carneiro. Análise de desempenho de métodos de acesso espaciais baseada em um banco de dados real. Master's thesis, Universidade Estadual de Campinas - UNICAMP, 1998.
- [CDK94] G. Couloris, J. Dollimore, and T. Kindberg. *Distributed Systems: Concepts and Design*. Addison-Wesley, 1994.
- [Cer96] N. Cereja. Visões em sistemas de informações geográficas - modelo e mecanismos. Master's thesis, UNICAMP, 1996.

- [Cha98] Ritu Chadha. Integration of web with legacy systems through java applets and distributed objects. In *Workshop on Compositional Software Architectures*, Monterey, California, January 1998.
- [Che76] P. Chen. The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9-36, 1976.
- [Che90] A. Chen. Outer join imization in multidatabase system. *2nd Int'l Symp. on Distributed and Parallel Database Systems*, pages 211-218, 1990.
- [CKRZ98] M. K. Chandy, J. Kiniry, A. Rifkin, and D. Zimmerman. A framework for structured distributed object computing. *Parallel Computing*, 24(12-13):1901-1922, 1998.
- [Cod70] E. F. Codd. A relational model for large shared data banks. *Communications of the ACM*, 13(6):377-387, 1970.
- [Cor01] Oracle Corporation. Oracle spatial. <http://www.oracle.com>, 2001.
- [Dav93] T. H. Davenport. *Processo Inovation, Reengineering Work through Information Technology*. MA:Harvard Business School Press, 1993.
- [DD96] C. J. Date and Hugh Darwen. *A Guide to SQL Standard*. Addison Wesley Longman, Inc., 1996.
- [DD99] C. J. J. Date and C. J. Date. *An Introduction to Database Systems*. Addison Wesley Longman, Inc., 1999.
- [DD00] C. J. Date and Hugh Darwen. *Foundation for Future Database Systems: The Third Manifesto*. Addison Wesley Longman, Inc., 2000.
- [DGMa95] E. Dias, S. Granado, and G. C. Magalhães. Uso de versões para garantia de consistência em ambientes híbridos de engenharia e operação. *Anais IX Simpósio Brasileiro de Banco de Dados*, Outubro 1995.
- [EN94] R. Elmarsri and S. Navathe. *Fundamentals of Database Systems*. The Benjamin/Cummings Publ Co., 1994.
- [FMa92] Cox F.S. and G.C Magalhães. Implementação e análise de métodos de acesso a dados espaciais. In *VII Simpósio Brasileiro de Banco de Dados*, Porto Alegre, Maio 1992. Addison-Wesley Publishing.
- [FS97] M. Fowler and K. Scott. *UML DISTILLED - applying the standard object modeling language*. ADDISON-WESLEY, 1997.

- [Gar95] D. Garlan. Research directions in software architecture. *ACM Computing Surveys*, 27(2):257–261, Junho 1995.
- [GB95] N. Ganti and W. Brayman. *The Transition of Legacy Systems to a Distributed Architecture*. John Wiley & Sons, Inc., 1995.
- [GEOG94] J. Gonçalves, M. Erhardt, F. Obata, and S. Granado. Automação de cadastro de rede de telecomunicações. *Anais do I Encontro de Qualidade de Rede e Equipamentos de Telecomunicações*, Setembro 1994.
- [GFKH99] A. Grimshaw, A. Ferrari, F. Knabe, and M. Hhumphrey. Wide-area computing: Resource sharing on a large scale. *Computer*, 32(5):29–37, May 1999.
- [GR93] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques*. Morgan-Kaufmann, 1993.
- [Gro93] Object Management Group. *Object Management Architecture Guide*. John Wiley & Sons, Inc., 1993.
- [HB91] A. R. Hurson and M. W Bright. Multidatabase systems: An advanced concept in handling distributed data. *Advances in Computers*, 32:149–200, 1991.
- [HBRY91] C. Hsu, M. Bouziane, L. Rattner, and L. Yee. Information resources management in heterogeneous, distributed environment: A metadatabase approach. *IEEE Transactions on Software Engineering*, 17(6):604–624, 1991.
- [HC93] M. Hammer and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business, 1993.
- [Inc98] SHL Systems Inc. *Vision Concepts*. SHL Systems Inc., 1998.
- [JEJ95] I. Jacobson, M. Ericsson, and A. Jacobson. *The Object Advantage - Business process reengineering with object technology*. Addison-Wesley Publishing, 1995.
- [Kam96] A. Kamada. Transformação de esquema relacional para esquema orientado a objetos em sistemas de bancos de dados heterogêneos. Master's thesis, UNICAMP, 1996.
- [Kim90] W. Kim. *Introduction to Object-Oriented Databases*. MIT Press, 1990.
- [LG01] X.R. Lopez and A. Gopalan. Managing long transactions using standard dbms technology. *Proceedings of the XXIV GITA Anual Conference*, March, 4-7 2001.

- [LPO97] J. C. D. Lima, Ribeiro C. P., and J. P. M. Oliveira. Acesso integrado a banco de dados distribuídos heterogêneos utilizando corba. *Anais do XII Simpósio de Banco de Dados*, 1997.
- [Ma97] G. C. Magalhães. Telecommunications outside plant management throughout brazil. *Proceedings of the XX International Conference on Automated Mapping / Facility Management*, Março 1997.
- [Ma00] G. C. Magalhães. Integrating operations and engineering gis. *GIS2000*, Outubro 2000.
- [Mer95] E. Merlo. Reengineering user interfaces. *IEEE Software*, 12(1), Janeiro 1995.
- [MM97] T. J. Mowbray and R. C. Malveau. *CORBA Design Patterns*. John Wiley & Sons, Inc., 1997.
- [MNB⁺94] L. Markosian, P. Newcomb, R. Brand, S. Burson, and T. Kitzmiller. Using an enabling technology to reengineer legacy systems. *Communications of the ACM*, 37(5), Maio 1994.
- [MS92] Jim Melton and A. R. Simon. *Understanding the New SQL: A Complete Guide*. Morgan-Kaufmann, October 1992.
- [NEK94] Jim Q. Ning, Andre Engsberts, and W Kozaczynski. Automated support for legacy code understanding. *Communications do the ACM*, 37(5), Maio 1994.
- [NM95] O. Nierstrasz and T. D. Meijler. Research directions in software composition. *ACM Computing Surveys*, 27(2):262–264, Junho 1995.
- [OCMa95] J. Oliveira, C. Cunha, and G. Magalhães. Modelo de objetos para construção de interfaces visuais dinâmicas. *Anais IX Simpósio Brasileiro de Engenharia de Software*, pages 143–158, Outubro 1995.
- [Oli93] R. L. Oliveira. Transparência de modelo em sistemas de banco de dados heterogêneo. Master's thesis, UNICAMP, julho 1993.
- [OMa93] R. L. Oliveira and G. C. Magalhães. Metodologia para conversão de esquemas em sistemas de bancos de dados heterogêneos. relatório técnico dcc-17/93. Technical report, UNICAMP, 1993.
- [OMG95] OMG. *Corbaservices: Common object services specification*, 1995.
- [OMG98a] OMG. *The common object request broker: Architecture and specification*, 1998.

- [OMG98b] OMG. Corba messaging specification, May 1998.
- [OOS01] O. Othman, C. O’Ryan, and D. Schmidt. The design of an adaptive corba load balancing service. *IEEE Distributed Systems Online*, 2(4), 2001.
- [PCC93] M.C. Paulk, M.B. Curtis, B.Chissis, and Weber C. Capability maturity model for software. Technical Report CMV/SEI-93-TR-024 ESC-TR-93-177, SEI, february 1993.
- [PM88] J. Peckham and R. Maryanski. Semantic data models. *ACM Computing Surveys*, 20(3):153–189, Setembro 1988.
- [PMa02] C.A. Previdelli and G.C. Magalhães. Moving geospatial applications towards a mission critical scenario. *to appear: Proceedings of the XXV GITA Anual Conference*, March, 17-20 2002.
- [PTO95] A. Prezzoto, D. Teijeiro, and F. Obata. Automação de projeto de rede externa de telecomunicações. *Anais II Congresso Internacional de Engenharia de Informação, Universidade de Buenos Aires, Argentina*, Novembro 1995.
- [RHH96] A. Rosenthal, R. Hyland, and E. Hughes. How to componentize data-intensive legacy applications: Issues and initial approaches. <http://www.mitre.org>, 1996.
- [RHH98] Arnon Rosenthal, Robert Hyland, and Eric Hughes. How to componentize data-intensive legacy applications: Issues and initial approaches. In *Workshop on Compositional Software Architectures*, Monterey, California, January 1998.
- [Riv97] E. Rivas. Enterprise component modeling strategies. *Object Magazine*, 7(2):62–64, April 1997.
- [Rob97] P. Robertson. Integration legacy systems with modern corporate applications. *Communications of the ACM*, 40(5):39–46, Maio 1997.
- [SB96] L. T. Sharon and M. Bitzer. Improving business process through the use of the workflow reengineering methodology. <http://hsb.baylor.edu/ramsower/ais.ac.96/papers/KAMEL.htm>, 1996.
- [SL90] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computing Surveys*, 22:183–236, Setembro 1990.
- [SM01] Inc. Sun Microsystems. <http://java.sun.com/>, 2001.

- [Soa98] H. R. Soares. Uma metodologia para integração de sistemas legados e bancos de dados heterogêneos. Master's thesis, UNICAMP, dezembro 1998.
- [SP94] S. Spaccapietra and C. Parent. View integration: A step forward in solving structural conflicts. *IEEE Transactions on Knowledge and Data Engineering*, 6(2):258-274, Abril 1994.
- [Spe89] A. Z. Spector. Distributed transaction processing facilities. In Sape Mullender, editor, *Distributed Systems*, pages 191-214. ACM Press, 1989.
- [SS96] A. Seidman and A. Sundararaja. Information technology, performance control & organizational structures: Effects on business process redesign. In *Proceedings of the Second AIS Americas Conference on Information Systems*, 1996.
- [Sto88] M. Stonebraker. Distributed databases systems introduction. In Michael Stonebraker, editor, *Readings in Databases Systems*, pages 189-195. Morgan Kaufman, 1988.
- [Sun96] A. Sundararaja. Business process design and organizational structure: Technological, operational and economic issues. In *Proceedings of the Second AIS Americas Conference on Information Systems*, 1996.
- [SV98] D.C. Schmidt and S. Vinoski. An introduction to corba messaging. *C++ Report*, 10(10), 1998.
- [SZ96] A. Silberschatz and S. Zdonik. Strategic direction in database systems - breaking out of the box. *ACM Computing Surveys*, 28(1), Março 1996.
- [TK78] D. Tsichritzis and A. Klug. The ansi/x3/sparc dbms framework. *Information Systems*, 3(4), 1978.
- [ULM97] E. M. A. Uchôa, S. Lifschitz, and R. N. Melo. Interoperabilidade em um sistema de bancos de dados heterogêneos usando padrão corba. In *Anais do XII Simpósio de Banco de Dados*, Outubro 1997.
- [VB97] V. Vidal and Lóscio B.F. Especificação de mediadores para acesso e atualização de múltiplas bases de dados. In *Anais do XII Simpósio de Banco de Dados*, Outubro 1997.
- [Weg96] P. Wegner. Interoperability. *ACM Computing Surveys*, 28(1), Março 1996.
- [Wie92] G. Wiederhold. Mediators in the architecture of future information systems. *Computer*, 25(3):38-49, março 1992.

- [Win95] Paul Winsberg. Legacy code: Don't bag it, wrap it. *Datamation*, Maio 1995.
- [WSM01] S. Wilson, H. Sayers, and M.D.J. McNeill. Using corba middleware to support the development of distributed virtual environment applications, 2001.
- [Yar97] P. J. Yarka. A recent technology direction: storing new data types in an rdbms. *Proceedings of the XX International Conference on Automated Mapping / Facility Management*, Março 1997.
- [Zha97] J. L. Zhao. Schema coordination in federated database management: a comparison with schema integration. *Decision Support Systems*, 20:243–257, 1997.