

Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2569

**DETALHAMENTO DO PASSO DE ABSTRAÇÃO
DO MODELO DE ANÁLISE DO MÉTODO
FUSION/RE**

**ROSANA TERESINHA VACCARE BRAGA
PAULO CESAR MASIERO**

Nº 70

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
Mar./1998

SYSNO	0948757
DATA	/ /
ICMC - SBAB	

Sumário

1 - INTRODUÇÃO.....	1
1.1 - CONSIDERAÇÕES INICIAIS.....	1
1.2 - TRABALHOS ANTERIORES RELEVANTES.....	1
1.3 - ORGANIZAÇÃO	2
2 – OS PASSOS DA ENGENHARIA REVERSA REALIZADA.....	2
2.1 - DESCRIÇÃO DA APLICAÇÃO	2
2.2 - PASSO 1: REVITALIZAÇÃO DA ARQUITETURA DO SISTEMA ATUAL.....	2
2.3 - PASSO 2: CRIAÇÃO DO MODELO DE ANÁLISE DO SISTEMA ATUAL (MASA).....	5
2.3.1 – <i>Modelo de Objetos do Sistema Atual</i>	5
2.3.2 – <i>Modelo de Ciclo de Vida do Sistema Atual</i>	8
2.3.3 – <i>Modelo de Operações do Sistema Atual</i>	10
2.4 - PASSO 3: CRIAÇÃO DO MODELO DE ANÁLISE DO SISTEMA (MAS).....	11
2.4.1 – <i>Modelo de Objetos do Sistema</i>	13
2.4.2 – <i>Modelo de Ciclo de Vida do Sistema</i>	22
2.4.3 – <i>Modelo de Operações do Sistema</i>	23
2.5 – PASSO 4: CORRESPONDÊNCIA ENTRE O MAS E O MASA	23
3 - CONCLUSÕES.....	24
3.1 - CONSIDERAÇÕES INICIAIS	24
3.2 – CONTRIBUIÇÕES DESTE TRABALHO	25
3.3 - CONCLUSÕES FINAIS.....	25
3.4 - SUGESTÕES PARA NOVOS TRABALHOS	25
BIBLIOGRAFIA.....	26
APÊNDICE A – ESQUEMAS PARA MAIS ALGUMAS OPERAÇÕES DO MOPSA.....	27
APÊNDICE B – ESQUEMAS PARA MAIS ALGUMAS OPERAÇÕES DO MOPS	29

Lista de Figuras e Tabelas

Figura 1 – Esquema da Base de Dados.	3
Figura 2 – Descrição de um dos arquivos da base de dados.	4
Figura 3 – Descrição de um dos módulos do sistema, com a relação chama/chamado por.	4
Figura 4 – MOSA – Modelo de Objetos do MASA	6
Figura 5 – MCVSA - Modelo de Ciclo de Vida do Sistema Atual	9
Figura 6 – Parte dos menus do sistema legado	10
Figura 7 – Esquema para a operação “lançamento_prim_parte” do MOpSA.....	11
Figura 8 - Trecho de um programa Clipper do sistema legado	12
Figura 9 – Processo de Transição do MASA para o MAS	12
Figura 10 – MOS-1 – Modelo de Objetos do MAS – Etapa 1	14
Figura 11 – MOS-2 – Modelo de Objetos do MAS – Etapa 2	15
Figura 12 – MOS-3 – Modelo de Objetos do MAS – Etapa 3	16
Figura 13 – MOS-4 – Modelo de Objetos do MAS – Etapa 4	18
Figura 14 – MOS-5 – Modelo de Objetos do MAS – Etapa 5	19
Figura 15 – MOS-6 – Modelo de Objetos do MAS – Etapa 6	20
Figura 16 – MOS-7 – Modelo de Objetos do MAS – Etapa 7	21
Figura 17 – MCVS - Modelo de Ciclo de Vida do Sistema	22
Figura 18 – Esquema para a operação “abre_ordem_serv” do MOpS.....	23
Tabela 1 – Correspondência entre uma classe do MAS e o arquivo do MASA	24
Tabela 2 – Correspondência entre algumas operações do MAS e do MASA.....	24

Siglas Utilizadas

MASA	Modelo de Análise do Sistema Atual
MAS	Modelo de Análise do Sistema
MOSA	Modelo de Objetos do MASA
MOS	Modelo de Objetos do MASA
MCVSA	Modelo de Ciclo de Vida do Sistema Atual
MCVS	Modelo de Ciclo de Vida do Sistema
MOpSA	Modelo de Operações do Sistema Atual
MOpS	Modelo de Operações do Sistema

1 - Introdução

1.1 - Considerações iniciais

Ao se realizar a engenharia reversa orientada a objetos de sistemas legados desenvolvidos com orientação a procedimentos, usando o Fusion/RE [Pen 96], sente-se a falta de uma sistemática que norteie a transição entre os modelos de análise, principalmente na fase de generalização do modelo de análise do sistema atual para um modelo de análise orientado a objetos.

Assim, percebe-se a utilidade de documentar cada etapa intermediária dessa fase, fazendo um trabalho preliminar que poderá ser aperfeiçoado em outras aplicações. Esse trabalho é feito aqui com um sistema legado desenvolvido em 1990 na linguagem Clipper, com cerca de 20.000 linhas de código, referente a um Sistema de Oficina Auto-elétrica e Mecânica.

1.2 - Trabalhos anteriores relevantes

O método Fusion para desenvolvimento orientado a objetos, proposto por Coleman [Col 94], baseia-se em diversos métodos orientados a objetos anteriores, como o OMT [Rum 91], Booch [Boo 91] e CRC [Bec 89]. Cobre todo o ciclo de desenvolvimento de um sistema, desde a análise, passando pelo projeto e até chegar à implementação.

Penteado, em [Pen 96], propõe um método de engenharia reversa orientada a objetos, baseado no método Fusion, constituído de quatro passos:

- 1- Revitalizar a Arquitetura do Sistema com base na documentação existente,
- 2- Recuperar o Modelo de Análise da Solução Atual,
- 3- Abstrair o Modelo de Análise do Sistema,
- 4- Mapear o Modelo de Análise do Sistema para o Modelo de Análise do Sistema Atual.

É justamente no passo 3 que as dificuldades mencionadas na seção 1.1 se apresentam. Esse passo tem por objetivo “obter um Modelo de Análise do Sistema considerando os aspectos do domínio da aplicação” e tem como subpassos :

- 1- Desenvolver o Modelo de Objetos,
- 2- Elaborar o Modelo de Ciclo de Vida,
- 3- Especificar o Modelo de Operações.

No subpasso 1, cuja meta é “elaborar um Modelo de Objetos considerando as classes e seus relacionamentos que devem ser tratados pelo sistema”, fica evidenciada a dificuldade de abstrair as classes e relacionamentos a partir do pseudo-modelo de objetos do sistema legado.

No trabalho de Penteado o método proposto é aplicado a um ambiente de software para projetar sistemas reativos com 30.000 linhas de código C. Esse ambiente usa *statecharts* para modelar o sistema em projeto e permite editar e simular *statecharts*. Nota-se que, nesse trabalho, a generalização do MASA para o MAS é apresentada como uma única transformação, o que não deixa suficientemente claras as diretrizes e heurísticas utilizadas.

Em [Pen 96a], o método desenvolvido em [Pen96] é denominado “Fusion/RE”, sendo

apresentado de forma resumida e sendo mostradas duas métricas para avaliação do esforço necessário para reengenharia. Em [Pen 96b] são discutidas as vantagens da aplicação do método Fusion/RE, tais como na melhoria da manutenibilidade atual e na facilidade de evolução do sistema por reengenharia. Alguns caminhos a serem seguidos após a aplicação da engenharia reversa são apresentados.

1.3 - Organização

Este trabalho está organizado da seguinte forma. No Capítulo 2 são descritos os passos da Engenharia Reversa realizada. Na seção 2.1 é feita a descrição da aplicação utilizada como exemplo. Na seção 2.2 descreve-se o trabalho de recuperação das informações sobre o sistema legado. Na seção 2.3 descreve-se a criação do modelo de análise do sistema atual. Na seção 2.4 detalha-se o processo de transição do modelo de análise do sistema atual para o modelo de análise do sistema. Na seção 2.5 faz-se a correspondência entre o MAS e o MASA. No Capítulo 3 são apresentadas as conclusões.

2 – Os passos da Engenharia Reversa realizada

2.1 - Descrição da aplicação

O sistema de Oficina Auto-elétrica e Mecânica utilizado como exemplo neste trabalho é basicamente um sistema de controle dos serviços executados em uma oficina de veículos, que faz também o controle de estoque das peças utilizadas. O cliente vai até a oficina para solicitar a realização de serviços em seu veículo, sendo que um cliente pode ter diversos veículos. O mesmo veículo pode voltar à oficina diversas vezes, sendo elaborada, em cada uma delas, uma Ordem de Serviço distinta. Essa Ordem de Serviço contém os dados do cliente, do veículo e dos reparos a serem feitos.

Quando o veículo é consertado, a Ordem de Serviço é complementada, introduzindo-se as peças utilizadas e a mão-de-obra executada. Muitas vezes o reparo pode exigir peças não existentes no estoque, que são adquiridas fora da oficina mecânica e também participam da Ordem de Serviço. O registro dessas peças é importante para o gerente da oficina, pois essas são candidatas a serem estocadas no futuro. Os possíveis modelos de veículos devem ser cadastrados pelo sistema, pois são utilizadas tabelas para cobrança de serviços, tanto elétricos como mecânicos, de acordo com o modelo de veículo.

2.2 - Passo 1: Revitalização da arquitetura do sistema atual

A revitalização da arquitetura do sistema consiste na recuperação de informações relacionadas à arquitetura do sistema, com base na documentação existente. O programa fonte, as estruturas de dados, os manuais do usuário e a própria execução do sistema são utilizados nesse passo.

O sistema em questão já possuía uma documentação razoável, constituída da descrição detalhada da base de dados [Oli 91], da relação “chama/chamado por” [Oli 91] e de um manual do usuário [Bra 90]. Algumas partes dessa documentação são mostradas nas figuras 1, 2 e 3.

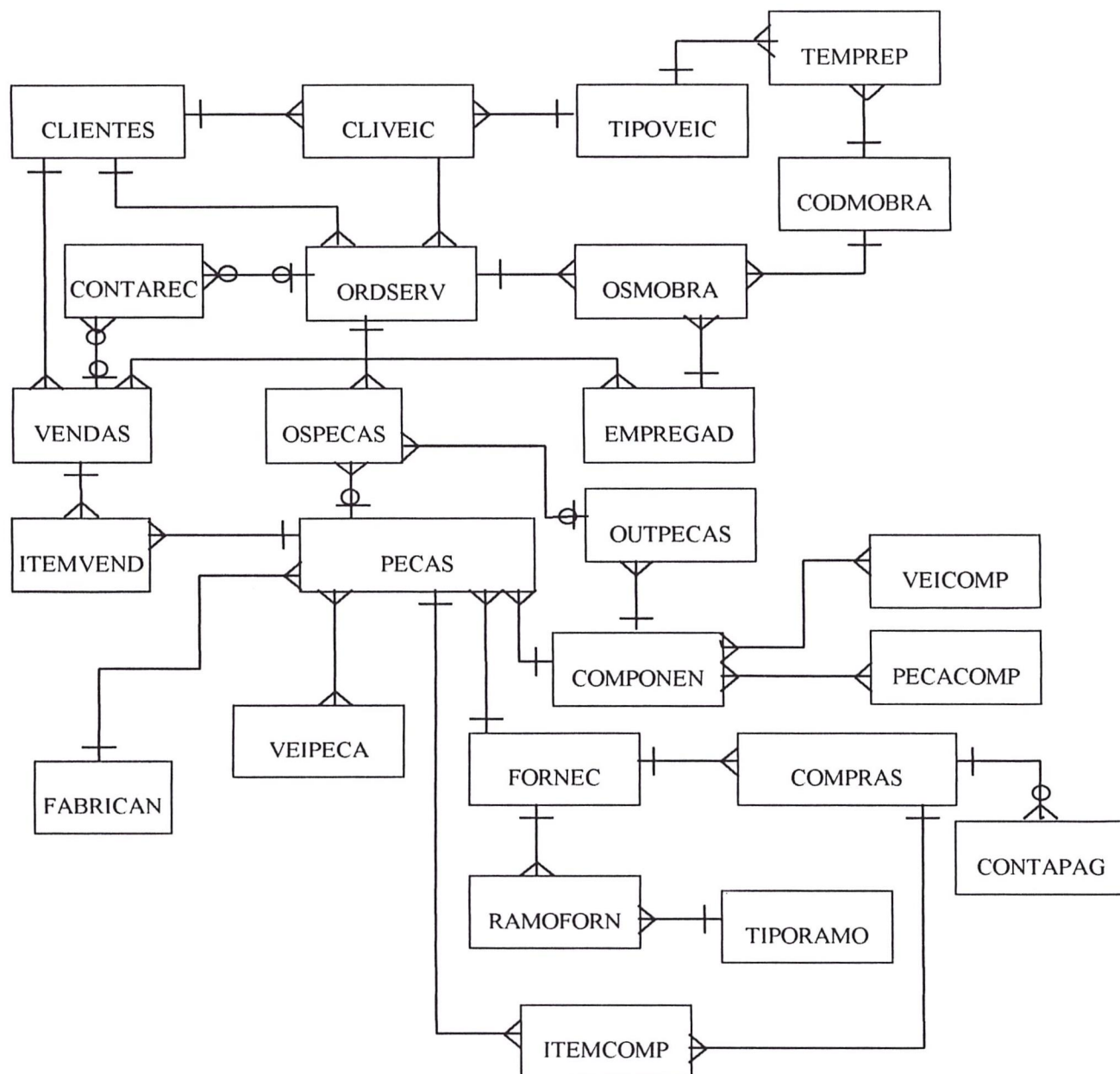


Figura 1 – Esquema da Base de Dados.

Porém, ao prosseguir com os demais passos, foram percebidas algumas falhas nessa documentação, tanto com referência aos arquivos da base de dados como à relação chama/chamado por. Por exemplo, umas das falhas foi encontrada nos arquivos “contas a receber” e “contas a pagar”, que constavam como dois arquivos na documentação, mas a implementação trabalhava com apenas um arquivo, “contas”, com um campo diferenciando o tipo de conta. Uma falha encontrada na relação “chama/chamado por” foi a presença apenas das chamadas a rotinas externas, ao invés de relacionar também chamadas a procedimentos internos.

+=====+				
ESTRUCTURA DO ARQUIVO ORDSERV .dbf				
+=====+				
ORDEM DE SERVIÇO				
+-----+				
Nro	Nome do	Tipo	Nro de	Descrição
Campo	Campo	do Campo	Caract.	
+=====+				
1	OSNRO	NUMERICO	5	NUMERO DA ORDEM DE SERVICO
2	DATAENTRA	DATA	8	DATA DE ENTRADA DA ORD. SERVICO
3	DATACOMPRA	DATA	8	DATA DA COMPRA DAS PECAS P/GAR.
4	CODCLIENTE	NUMERICO	5,0	CODIGO DO CLIENTE
5	NROVEICULO	NUMERICO	4,0	NUMERO DO VEICULO DO CLIENTE
6	SERVAEX1	CARACTER	60	SERVICOS A EXECUTAR - PARTE 1
7	SERVAEX2	CARACTER	60	SERVICOS A EXECUTAR - PARTE 2
8	SERVAEX3	CARACTER	60	SERVICOS A EXECUTAR - PARTE 3
9	ORCAMENTO	CARACTER	1	INDICADOR DO ORCAMENTO (S/N)
10	DATAPROM	DATA	8	DATA PROMETIDA PARA ENTREGA
11	HORAPROM	DATA	8	HORA PROMETIDA PARA ENTREGA
12	CONDPAG	CARACTER	15	CONDICOES DE PAGAMENTO
13	PRECOTOT	NUMERICO	11,2	PRECO TOTAL COBRADO
14	DESCONTO	NUMERICO	11,2	DESCONTO CONCEDIDO
15	DESPEXTRA	NUMERICO	11,2	DESPESAS.EXTRAS INCLUSAS
+-----+				

ARQUIVOS DE INDICE

CLIORDSE.ntx --> CODCLIENTE

ORDSERV .ntx --> OSNRO

Programas que o leem:

ALTCAREC ELICAREC INDEXACA INSCAREC QUITCARE

Programas que o alteram:

OSTELA1 OSTELA2

Figura 2 – Descrição de um dos arquivos da base de dados.

```

-----
Módulo Físico:  OSTELA1
Descrição: PRIMEIRA TELA DA ORDEM DE SERVICO
Arquivos apenas lidos:
Arquivos lidos e/ou alterados:  CLIENTES  CLIVEIC  ORDSERV  TIPOVEIC
Rotinas chamadas:  LOGOTIPO
Rotinas que o chamam:  LANCAMEN
-----

```

Figura 3 – Descrição de um dos módulos do sistema, com a relação chama/chamado por.

2.3 - Passo 2: Criação do Modelo de Análise do Sistema Atual (MASA)

2.3.1 – Modelo de Objetos do Sistema Atual

O diagrama Entidade-Relacionamento da base de dados (figura 1), embora contivesse alguns erros, foi útil na construção do modelo de objetos do MASA, denominado de Modelo de Objetos do Sistema Atual (MOSA - figura 4), que foi elaborado seguindo a notação do método Fusion [Col 94]. Em princípio, os relacionamentos foram denotados por losangos desenhados com linhas tracejadas, significando que são, na verdade, pseudo-relacionamentos, já que são implementados por chaves estrangeiras nas tabelas relacionais. Nota-se que não foram atribuídos nomes significativos a esses relacionamentos, pois para isso seria necessário um maior conhecimento sobre o domínio de aplicação.

Foram incluídos no Modelo de Objetos apenas alguns atributos de cada classe, suficientes para que se tenha uma idéia do seu significado. O campo-chave de cada classe foi colocado em negrito e sempre no início da lista, enquanto que as chaves estrangeiras foram sublinhadas. Por exemplo, a classe *Clientes* relaciona-se com a classe *Cliveic* pelo campo-chave “**codcliente**”, que foi colocado em negrito em *Clientes* e sublinhado em *Cliveic*. Na classe *Tipoveic* foi utilizada uma linha vertical adicional, para denotar a duplicação dessa classe em outro local do Modelo de Objetos.

A classe *Clientes* tem como objetos todos os clientes oficina auto-elétrica e mecânica, inclusive um de particular interesse, que é o “consumidor”. Alguns veículos são submetidos a serviços rápidos, como por exemplo a carga de bateria, e não são cadastrados por completo. Nesse caso, é utilizado o cliente “consumidor”. Cada cliente da empresa possui um código de identificação único.

A classe *Cliveic* tem como objetos todos os veículos que passam pela empresa, seja para serviços elétricos ou mecânicos, inclusive para serviços rápidos. Na maioria das vezes, a identificação do veículo é feita por sua placa. Porém, para serviços rápidos, a placa do veículo não é anotada. Nesses casos, é atribuído um número sequencial ao veículo, e “consumidor” ao nome do cliente.

A classe *Tipoveic* tem como objetos todos os possíveis tipos de veículos existentes no mercado, seja de automóveis, motocicletas, caminhões, camionetes, etc. Para cada marca e modelo de veículo, é atribuído um código de identificação único.

Cada objeto da classe *Ordserv* representa uma ordem de serviço encaminhada aos empregados da empresa, dando-lhes ordens para efetuar serviços em um determinado veículo. Esses serviços incluem tanto mão-de-obra elétrica quanto mecânica, podendo ocorrer os dois tipos simultaneamente. A cada ordem de serviço é atribuído um número único.

A classe *Pecas* engloba todas as peças estocadas pela empresa, sendo que o identificador de cada objeto da classe é o código de barras impresso em cada peça. A quantidade em estoque de cada peça é de suma importância para a gerência, havendo uma quantidade mínima a partir da qual a peça passa a aparecer em um relatório de estoque crítico.

A classe *Outpeças* tem como objetos todas as peças adquiridas fora da empresa especialmente para serem usadas em uma ordem de serviço. É de interesse da gerência o controle dessas peças, que são candidatas a serem estocadas no futuro. O identificador para essas peças é o código do fabricante ou, na sua ausência, um código sequencial fornecido pelo sistema. Essa aquisição de peças não é tratada pelo sistema, que só cuida da reposição de estoque.

A classe *Ospeca* engloba todas as peças utilizadas em cada ordem de serviço, tendo como atributos a quantidade e preço cobrado na ocasião. A classe *Osmobra* engloba toda a mão-de-obra envolvida na sua execução, tendo como atributo o preço de cada uma delas.

A classe *Codmobra* tem como objetos todos os possíveis tipos de mão-de-obra a ser executada em veículos, tanto do tipo elétrico quanto mecânico. Cada tipo de mão-de-obra possui um código identificador único.

A classe *Temprep* engloba os tempos de reparo necessários à execução de determinados serviços de mão-de-obra, de acordo com o tipo de veículo. A classe *Vtrep* engloba as quantidades de valores de trabalho cobradas na execução de determinados serviços de mão-de-obra, de acordo com o tipo de veículo.

A classe *Empregad* tem como objetos todos os empregados da empresa, sendo que podem ser especializados na parte elétrica ou mecânica dos veículos. Os empregados recebem comissão pelos serviços executados.

A classe *Vendas* tem como objetos todos os atendimentos efetuados no balcão da empresa, sem envolver o veículo do cliente, ou seja, é efetuada apenas a venda de peças em estoque. O sistema gera um número sequencial único para cada atendimento efetuado. A classe *Itemvend* engloba todas as peças vendidas em um determinado atendimento por balcão, registrando a quantidade e preço unitário na ocasião da venda.

A classe *Fabrican* tem como objetos todos os fabricantes de peças para veículos. Cada um deles possui um código de identificação único. A classe *Equivpec* engloba todas as peças equivalentes a uma determinada peça.

A classe *Componen* tem como objetos componentes elétricos ou mecânicos, que na verdade são peças maiores, compostas de outras peças. Os componentes podem ser vendidos ou usados em ordens de serviço, sendo de interesse da gerência saber da sua composição para o caso de substituição apenas de partes do mesmo. Cada componente possui um código único de identificação.

A classe *Pecacomp* engloba todas as peças que fazem parte de um componente maior. A classe *Veicomp* engloba todos os veículos aos quais um determinado componente se adapta. A classe *Veipeca* engloba todos os veículos aos quais uma determinada peça se adapta.

A classe *Compras* tem como objetos as compras ou pedidos efetuados pela oficina auto-elétrica e mecânica a seus fornecedores de peças. A compra pode ser efetuada diretamente, ocasião em que as peças são acrescentadas ao estoque, ou pode haver um pedido prévio. No caso de pedido prévio, a gerência deve aguardar sua chegada e conferi-lo, dando baixa no pedido e acrescentando ao estoque apenas as peças que realmente chegaram. Um

relatório de crítica relaciona as diferenças entre a remessa e o pedido efetuado. O gerente da auto-elétrica e mecânica fica encarregado de fazer um novo pedido com base no relatório de crítica, pois o sistema elimina o pedido por completo. A classe *Itemcomp* engloba as peças compradas ou pedidas em uma determinada compra/pedido, armazenando a quantidade e o preço de ocasião.

A classe *Fornec* tem como objetos os fornecedores de peças para a empresa. Cada um deles tem um código identificador único. A classe *Ramoforn* engloba os ramos com os quais os fornecedores trabalham. A classe *Ramo* tem como objetos todos os possíveis ramos de negócios. Cada ramo tem um código identificador único.

A classe *Contarec* tem como objetos as contas a receber pela empresa, provenientes de atendimentos por balcão ou de ordens de serviço realizadas. A classe *Contapag* tem como objetos as contas a pagar pela empresa, referentes a compras efetuadas nos fornecedores.

2.3.2 – Modelo de Ciclo de Vida do Sistema Atual

O Modelo de Ciclo de Vida do Sistema Atual (MCVSA - figura 5) retrata o comportamento global do sistema, preocupando-se com a ordem cronológica em que as operações são realizadas. Foi construído baseado nas opções encontradas nos menus do sistema legado, que encontram-se detalhados em [Oli 91] e são parcialmente ilustrados na figura 6.

Observando a figura 5 nota-se que, em um sistema de informação tradicional, não existem tantas restrições quanto à ordem em que as operações são executadas. Como foi considerado o Modelo de Ciclo de Vida do Sistema, e não o Modelo de Ciclo de Vida de cada objeto, a execução das operações pode ser alternada de forma livre, por exemplo pode-se executar a operação “lançamento_prim_parte” diversas vezes, executar então a operação “lançamento_Seg_parte”, voltar a executar a operação “lançamento_prim_parte”, etc. Isso ilustra a realidade da Oficina Auto-elétrica e Mecânica, onde podem entrar diversos veículos um seguido do outro, apenas um deles ser consertado, entrar mais um, etc.

Por outro lado, se fosse construído o Modelo de Ciclo de Vida para apenas um objeto, haveria necessidade de estabelecer a ordem cronológica das operações. Por exemplo, o veículo chega à oficina, depois é consertado, depois são lançadas as contas a receber referentes ao conserto, que depois são quitadas. Poderiam ser encaixadas nesse intervalo de tempo as operações de alteração e cancelamento da ordem de serviço e das contas a receber, nos locais onde fosse apropriado.

lifecycle Auto-elétrico e Mecânica:= (Lançamentos | Relatórios | Consultas |
Manutenção_de_arquivos | Manutenção_de_tabelas | Impressos)*

Lançamentos = (lancamento_prim_parte . [#Ordem_Serviço_impressa]) |
lançamento_seg_parte . [#Relat_serviços_executados]) |
modificação_prim_parte | modificação_seg_parte | cancelamento_os |
inserção_pedido | modificação_pedido | eliminação_pedido |
inserção_compra | modificação_compra | eliminação_compra |
inserção_vendas | modificação_vendas | eliminação_vendas |
inserção_conta_a_pagar | alteração_conta_a_pagar |
eliminação_conta_a_pagar | quitação_conta_a_pagar |
inserção_conta_a_receber | alteração_conta_a_receber |
eliminação_conta_a_receber | quitação_conta_a_receber

Relatórios = (rel_ord_serv_pendentes | rel_ord_serv_do_mês | rel_pecas_estoque_critico
| rel_pecas_em_estoque | rel_precos_por_fabricante | rel_equivalencia_entre_pecas
| rel_pecas_por_componente | rel_pecas_adquiridas_fora | rel_pedidos_pendentes
| rel_pedidos_pendentes_por_fornecedor | rel_compras_mês | rel_itens_cancelados
| rel_vendas_mês | rel_contas_a_pagar | rel_contas_a_receber | rel_balanco_mensal
| rel_cliente_ord_alfabet | rel_cliente_por_entidade | rel_fornecedor_ord_alfabet
| rel_fornecedor_por_ramo | rel_fabricante | rel_empregado | rel_componentes
| rel_veiculos | rel_mao_de_obra | tabela_sugestao_precos
| tabela_servicos_eletricos | tabela_servicos_mecanicos

Consulta = (cons_peca_letra_inicial | cons_peca_fabricante | cons_peca_codigo_barras |
cons_peca_codigo_original | cons_equivalencia_peca | cons_estatistica_custos |
cons_fornecedores | cons_vendas | cons_empregados | cons_compras |
cons_componentes | cons_clientes)

Manutenção_de_arquivos = (inserção_pecas | alteração_pecas | eliminação_pecas |
inserção_fornecedores | alteração_fornecedores | eliminação_fornecedores |
inserção_empregados | alteração_empregados | eliminação_empregados |
inserção_clientes | alteração_clientes | eliminação_clientes |
inserção_fabricantes | alteração_fabricantes | eliminação_fabricantes |
inserção_componentes | alteração_componentes | eliminação_componentes |
inserção_mão_de_obra | alteração_mão_de_obra | eliminação_mão_de_obra |
inserção_veículos | alteração_veículos | eliminação_veículos |

Manutenção_de_tabelas = tabela_sugestão_de_preços | inserção_tabela_valor_trabalho |
alteração_tabela_valor_trabalho | eliminação_tabela_valor_trabalho |
inserção_tabela_tempo_reparo | alteração_tabela_tempo_reparo |
eliminação_tabela_tempo_reparo

Impressos = cheques | duplicatas | ord_servico | termo_garantia
| recibo | cancelamento_protestos | carta_cancelamento | vales
| etiq_cod_barra_por_fornecedor | etiq_cod_barra_por_peca
| etiq_cod_barra_por_compra | etiq_cod_barra_pecas_miudas | etiq_por_estantes
| etiq_clientes | etiq_pessoa_juridica | etiq_pessoa_fisica | etiq_cli_por_entidade
| etiq_clientes_desejados | etiq_clientes_duplicatas | etiq_fornecedores)

Figura 5 – MCVSA - Modelo de Ciclo de Vida do Sistema Atual

I-LANÇAMENTOS =====	{LANCAMEN}
1-ORDEM DE SERVIÇO	{MENUOS}
1-LANÇAMENTO 1ª PARTE	{OSTELA1}
2-LANÇAMENTO 2ª PARTE	{OSTELA2}
3-LANÇAMENTO TOTAL	{OSTELA}
4-MODIFICAÇÃO 1ª PARTE	{MODOS1}
5-MODIFICAÇÃO 2ª PARTE	{MODOS2}
6-CANCELAMENTO DA O.S.	{CANCELOS}
2-COMPRAS	{MENUCOM}
1-PEDIDO DE PEÇAS	{MENUPP}
1-INSERÇÃO	{INSEPEPE}
2-MODIFICAÇÃO	{ALTPEPE}
3-ELIMINAÇÃO	{ELIPEPE}
2-COMPRAS DE PEÇAS	{MENCUC}
1-INSERÇÃO	{INSECE}
2-MODIFICAÇÃO	{ALTPCE}
3-ELIMINAÇÃO	{ELICE}
3-VENDAS DE PEÇAS POR BALCÃO	{MENVEND}
1-INSERÇÃO	{INSECV}
2-MODIFICAÇÃO	{ALTCVE}
3-ELIMINAÇÃO	{ELICVE}
4-CONTAS A PAGAR	{CONTASPA}
1-INSERÇÃO	{INSCAPAG}
2-MODIFICAÇÃO	{ALTCAPAG}
3-ELIMINAÇÃO	{ELICAPAG}
4-QUITACÃO	{QUITCAPG}
5-CONTAS A RECEBER	{CONTASPA}
1-INSERÇÃO	{INSCAREC}
2-MODIFICAÇÃO	{ALTCAREC}
3-ELIMINAÇÃO	{ELICAREC}
4-QUITACÃO	{QUITCARE}

Figura 6 – Parte dos menus do sistema legado

2.3.3 – Modelo de Operações do Sistema Atual

Para cada operação definida no Modelo de Ciclo de Vida é construído o esquema correspondente, no qual ela é especificada.

A figura 7 exemplifica o Modelo de Operações do Sistema Atual (MOpSA), mostrando o esquema para o operação “lançamento_prim_parte”. O Apêndice A traz os esquemas de mais algumas operações consideradas importantes para o entendimento da lógica do sistema.

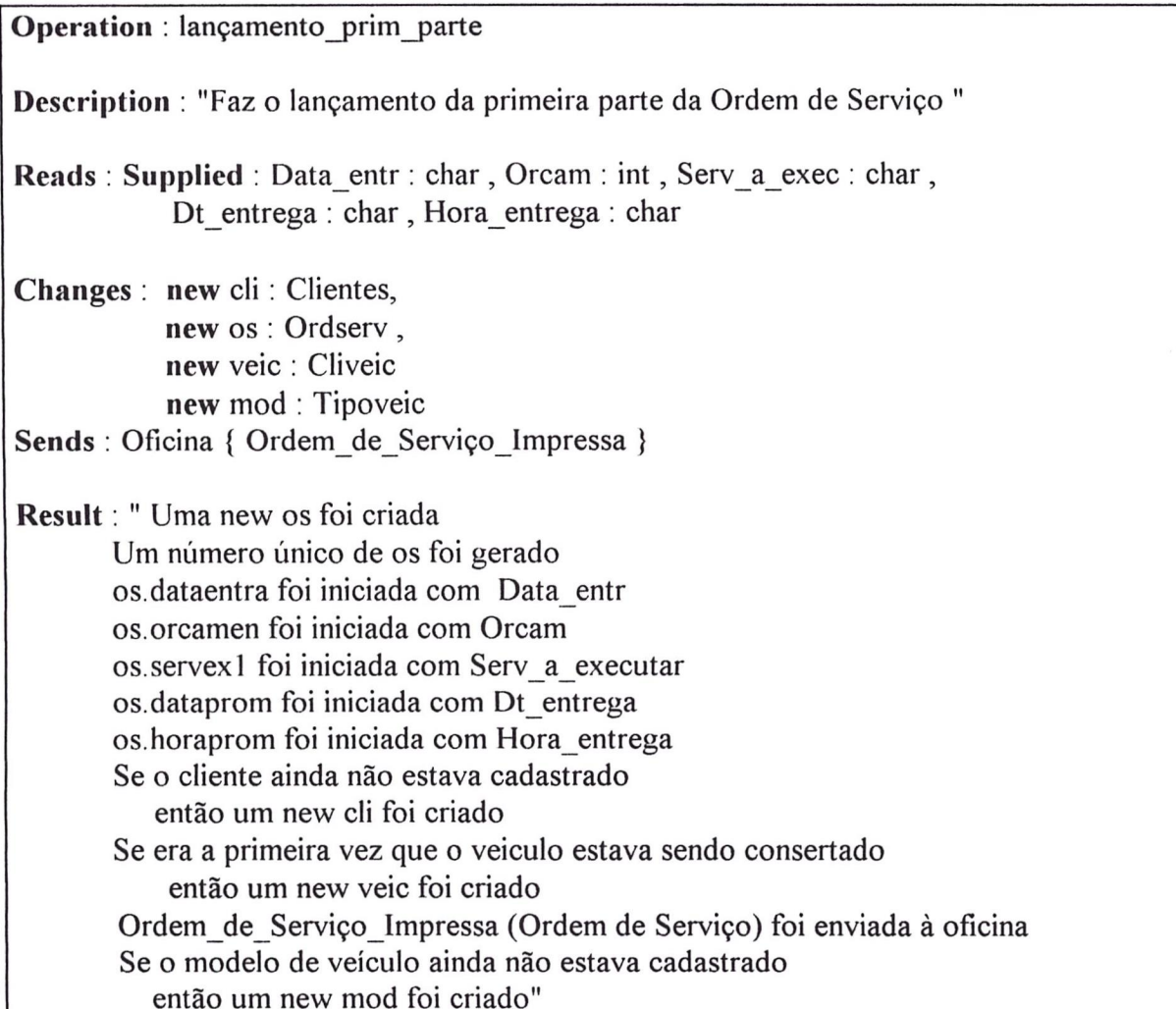


Figura 7 – Esquema para a operação “lançamento_prim_parte” do MOpSA

As informações necessárias para construir o Modelo de Operações foram obtidas tanto de forma estática (pelo exame de código-fonte) como de forma dinâmica (pela execução da opção correspondente à operação nos menus do sistema legado). A figura 8 mostra um trecho de um programa-fonte do sistema.

2.4 - Passo 3: Criação do Modelo de Análise do Sistema (MAS)

O Modelo de Objetos, o Modelo de Ciclo de Vida e o Modelo de Operações do MASA, vistos na seção anterior, serviram de base para a elaboração dos respectivos modelos do MAS, mostrados nas sub-seções que se seguem. A figura 9 ilustra esse processo de transição do MASA para o MAS.

Como pode ser observado na figura 9, a obtenção do Modelo de Objetos do MAS foi conseguida de forma gradual, elaborando-se diversos modelos (MOS-1 a MOS-7), cada um deles evoluindo a partir do anterior e acrescentando um novo aspecto ou fazendo uma abstração maior.

<pre> * Nome do programa:OSTELA1 SELECT 1 USE CLIENTES INDEX CODCLI, NOMECLI SELECT 2 USE ORDSERV INDEX CLIORDSE, ORDSERV SELECT 3 USE CLIVEIC INDEX CLICODVE SELECT 4 USE TIPOVEIC INDEX CODVEIC, MARCVEI, MARCTIP DO LOGOTIPO SELECT 2 SET ORDER TO 2 GO BOTTOM gravou=.f. codcli = " " nrov = 0 imp = " " nrord = OSNRO + 1 dtent = DATE() mv = SPACE(15) tv = SPACE(30) @ 6, 15 SAY "Lancamento da 1a. Parte da Ordem de Servico" @ 7, 15 SAY REPLICATE(' ', 40) @ 9, 5 SAY "Data de Entrada:" @ 9, 22 GET dtent PICTURE "@E" @ 9, 48 SAY "O.S.Nro:" @ 9, 56 SAY nrord PICTURE[99999] @ 10, 5 SAY "Codigo do Cliente:" @ 10, 24 GET codcli PICTURE [99999] when buscli() DO AVISO WITH " <99999>- Cliente nao Cadastrado <0> Abandonar" READ codcli=val(codcli) IF lastkey()#27 .and. codcli <> 0 </pre>	<pre> nro_os=nrord SELECT CLIENTES SET ORDER TO 1 SEEK codcli IF .NOT. FOUND() GO BOTTOM codcli = CODCLIENTE + 1 dtmov = DATE() APPEND BLANK REPLACE CODCLIENTE WITH codcli,DATAMOV WITH dtmov DO AVISO WITH "Digite os Dados do Novo Cliente" novo = "S" ELSE DO AVISO WITH "Verifique os Dados do Cliente, Modificando se Necessario" novo = "N" ENDIF ok = "N" DO WHILE ok = "N" @ 11, 5 SAY "Cliente:" IF novo = "S" @ 11, 14 GET NOME PICTURE REPL ("!",30) ELSE @ 11, 14 SAY NOME PICTURE REPL ("!",30) ENDIF @ 12, 5 SAY "Endereco:" @ 12, 15 GET ENDERECO PICTURE REPLICATE ("!",30) @ 12, 53 SAY "Bairro:" @ 12, 62 GET BAIRRO PICTURE REPLICATE ("!",15) . . . @ 20, 5 SAY "Entidade a qual Pertence:" @ 20, 33 GET ENTIDADE PICTURE REPLICATE ("!",20) READ NOME R=NOME DO CONFIRMA WITH "Corrigir algo", ok DO TELALIM ENDDO . . . </pre>
---	--

Figura 8 - Trecho de um programa Clipper do sistema legado

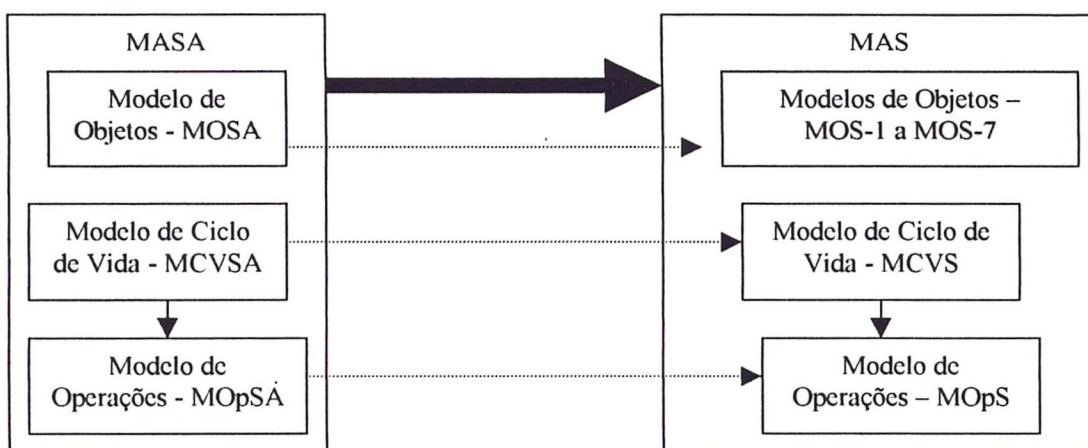


Figura 9 – Processo de Transição do MASA para o MAS

2.4.1 – Modelo de Objetos do Sistema

O Modelo de Objetos do MASA, mostrado na figura 4, serviu como base para a elaboração do Modelo de Objetos do MAS, que foi feito de forma evolutiva, mostrando em detalhes cada modificação incorporada, de forma a sistematizar o processo de generalização. Dessa forma, foram obtidos sete modelos de objetos, nomeados MOS-1 a MOS-7. O modelo de objetos final é constituído dos dois últimos (MOS-6 e MOS-7).

2.4.1.1 – MOS-1

Em uma primeira etapa foi elaborado o MOS-1, mostrado na figura 10. Os pseudo-relacionamentos existentes no MASA foram transformados em relacionamentos, removendo-se os atributos das classes cujo único objetivo era fazer o relacionamento entre tabelas. Ou seja, as chaves estrangeiras (atributos sublinhados) foram removidas e as linhas tracejadas tornaram-se contínuas. Por exemplo, da classe “Cliveic” foram removidos os atributos “codcliente” e “codveiculo”, cuja função era relacioná-la às classes “Clientes” e “Tipoveic”, respectivamente. Além disso, o relacionamento “rel-1”, que era tracejado, tornou-se contínuo.

2.4.1.2 – MOS-2

Em uma segunda etapa foi elaborado o MOS-2, mostrado na figura 11. Nele algumas tabelas do MASA foram transformadas em relacionamentos. Considerou-se que classes do MOS-1 que não continham atributos, ou que continham apenas um atributo fraco, poderiam ser transformadas em um relacionamento. Por exemplo, consideramos como atributo fraco o tempo gasto na execução de um determinado serviço, pois não identifica unicamente o tipo de mão-de-obra, já que podem existir diversos tipos de mão-de-obra com o mesmo tempo de execução.

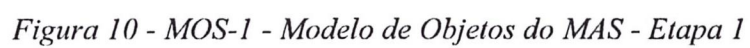
Dessa forma, foram tratadas as classes *Ramoforn*, *Temprep*, *Vtrep*, *Pecacomp*, *Veicomp* e *Veipeca*. Deve-se notar que, onde havia dois relacionamentos do tipo “um para muitos” foi obtido um relacionamento do tipo “muitos para muitos”. Alguns relacionamentos obtidos contêm atributos, como por exemplo o relacionamento “possui tempo de reparo de acordo com”, que tem o atributo “tempo”.

A classe *Equivpec* foi eliminada com base no conhecimento sobre o domínio da aplicação. A tabela “Equivpec” possui atributos “codpeca1” e “codpeca2”. Pode-se deduzir que referem-se a “codigo da peça 1” e “código da peça 2”, ou seja, a peça cujo código é “codpeca1” é equivalente à peça cujo código é “codpeca2”. Dessa forma, conclui-se que há um auto-relacionamento na classe “Peça”, denotando a equivalência entre duas ou mais peças.

2.4.1.3 – MOS-3

Em uma terceira etapa foi elaborado o MOS-3. Nessa etapa foram identificadas algumas agregações e especializações de classes, mostradas na figura 12.

Criaram-se agregações envolvendo as classes *Ordserv*, *Ospecas* e *Osmobra*, as classes *Vendas* e *Itemvend* e as classes *Compras* e *Itemcomp*. As especializações foram identificadas em classes com diversos atributos em comum, como é o caso das classes *Pecas* e *Outpecas* e das classes *Contarec* e *Contapag*.



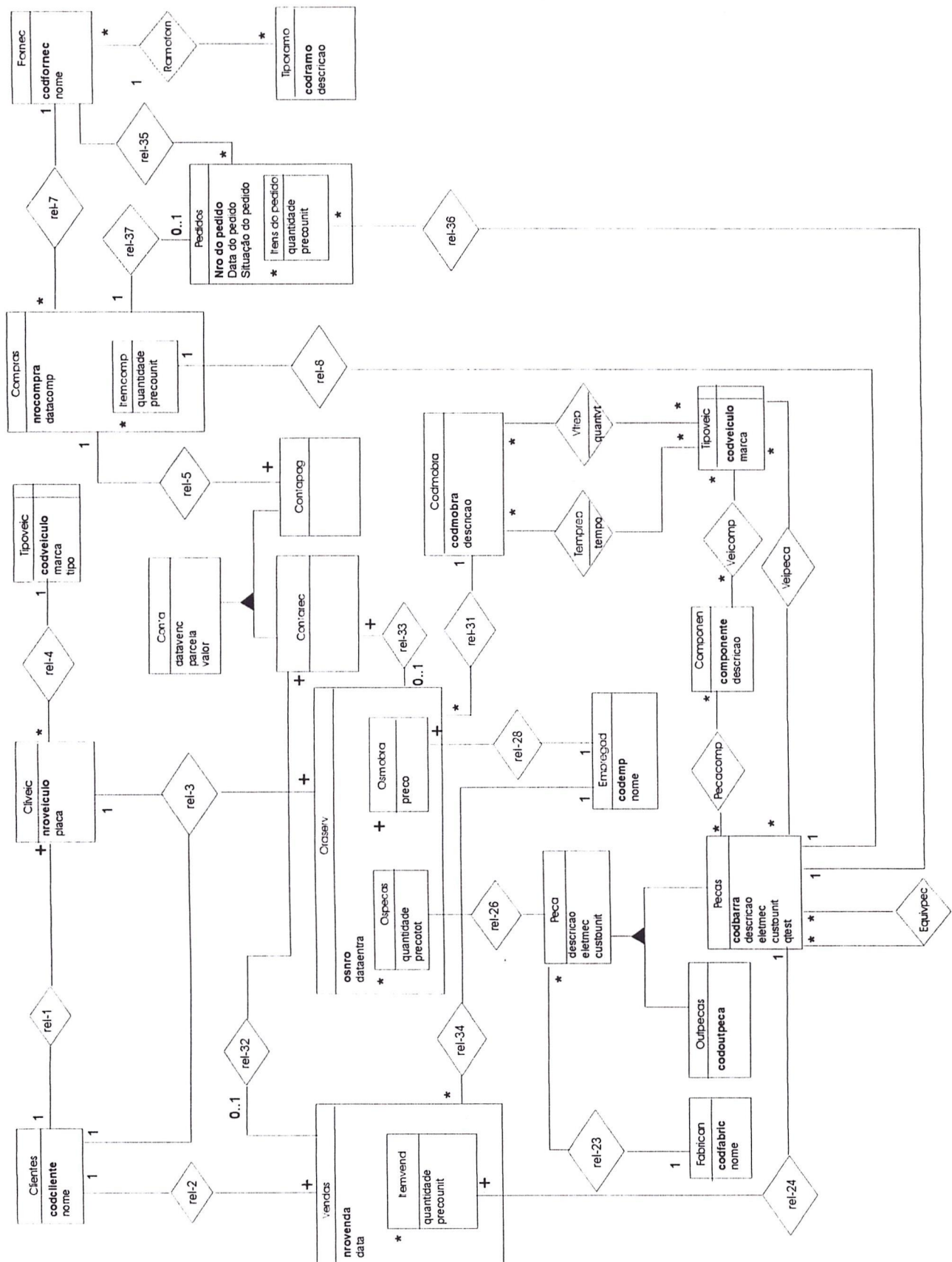


Figura 12 - MOS-3 - Modelo de Objetos MAS - Etapa 3

Em uma solução alternativa, as classes agregadas poderiam ter sido transformadas em relacionamentos, como no MOS-2, sendo que os atributos das classes tornar-se-iam atributos dos relacionamentos resultantes. Por exemplo, a classe *Itemcomp*, do MOS-1, poderia ser transformada em um relacionamento do tipo muitos para muitos entre a classe *Compra* e a classe *Peca*, com os atributos *quantidade* e *preco unit*, eliminando-se os relacionamentos “rel-6” e “rel-8” do MOS-1.

2.4.1.4 – MOS-4

Em uma quarta etapa foi elaborado o MOS-4, mostrado na figura 13. Nele foram acrescentadas algumas entradas mais relevantes do sistema e os agentes externos que interagem com o sistema, estabelecendo seus limites. Esses agentes são o atendente e o gerente da empresa. O atendente é necessário porque o sistema não é do tipo “self-service”. Existe um atendente intermediário, que faz os lançamentos no computador e recebe alguns relatórios de saída. Outros relatórios mais sofisticados são entregues ao gerente, que os analisa e toma suas decisões.

Como existem muitas operações nesse sistema, deve-se definir critérios para explicitação dos agentes externos e operações. Por exemplo, é claro que convém explicitar o papel do atendente de abrir e fechar uma ordem de serviço. Todavia, ao surgir um novo tipo de veículo há necessidade de incluir tempo de reparo e/ou valor de trabalho correspondente, o que seria feito pelo gerente. O mesmo aconteceria quando aparecesse um fornecedor novo, um empregado novo, etc.

2.4.1.5 – MOS-5

Em uma quinta etapa foi elaborado o MOS-5, mostrado na figura 14. Esse Modelo de Objetos mostra algumas saídas do sistema, bem como os agentes externos que as manuseiam. Por exemplo, o gerente acompanha o relatório de ordens de serviços pendentes e o relatório de peças com estoque crítico, o atendente utiliza a relação de clientes e o mecânico recebe a ordem de serviço impressa.

2.4.1.6 – MOS-6 e MOS-7

Nas etapas 6 e 7 foram elaborados o MOS-6 e MOS-7, mostrados nas figuras 15 e 16 respectivamente. No MOS-6 generalizaram-se os nomes de classes, atributos e relacionamentos referentes ao MOS-4, enquanto que no MOS-7 generalizaram-se os nomes de classes, atributos e relacionamentos do MOS-5. O objetivo foi de dar nomes mais significativos, de acordo com o domínio de aplicação. Também foram acrescentados os outros atributos das classes, que anteriormente haviam sido omitidos para facilitar o entendimento do modelo.

2.4.1.7 - Considerações

O Modelo de Objetos final do MAS é aquele constituído do MOS-6 e MOS-7. Os demais são modelos intermediários, que serviram apenas para detalhar o processo de transição do MASA para o MAS.

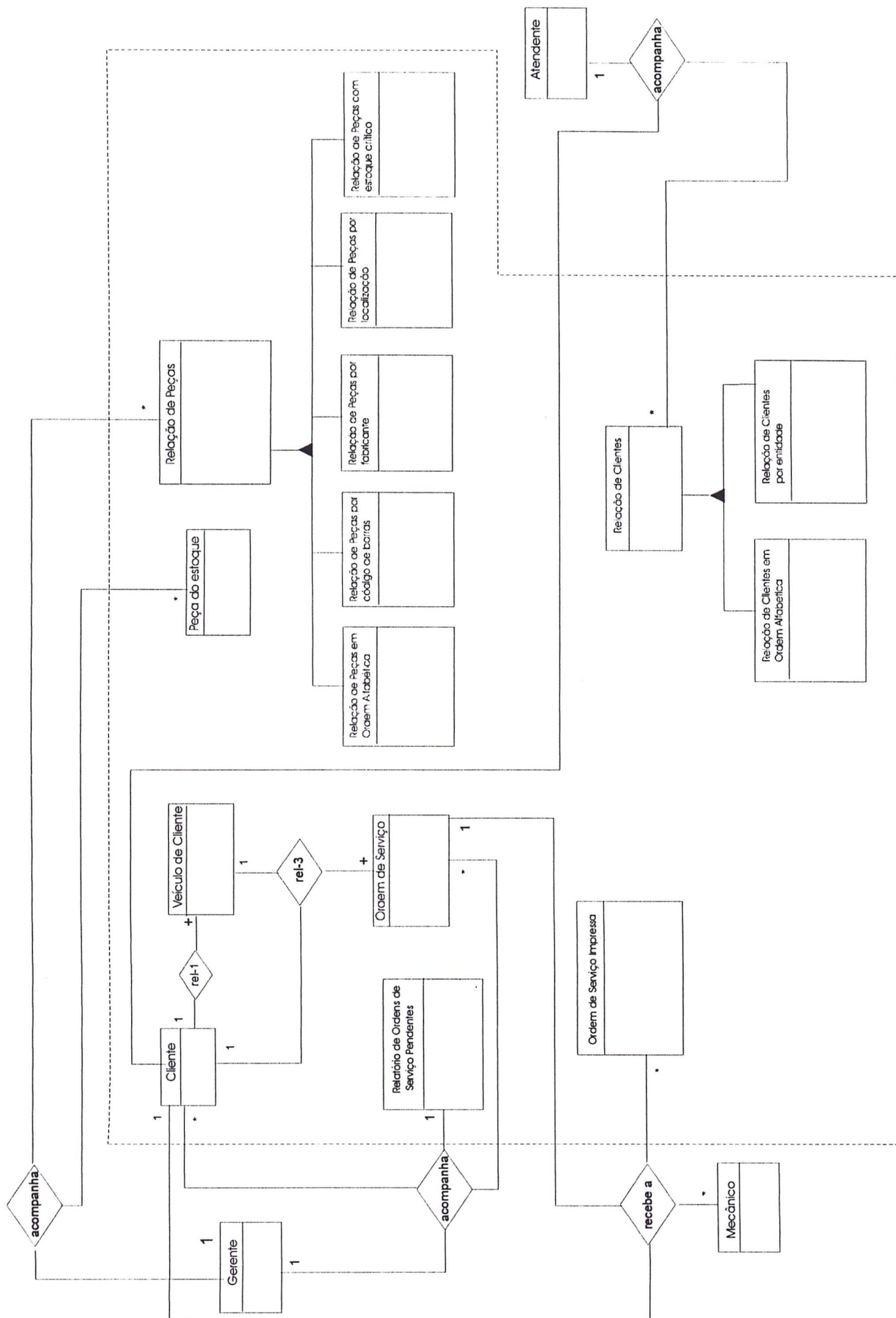


Figura 16 - MOS-7 - Modelo de Objetos do MAS - Etapa 7

2.4.2 – Modelo de Ciclo de Vida do Sistema

O Modelo de Ciclo de Vida do MAS (MCVS - figura 17) foi feito com base no Modelo de Ciclo de Vida do MASA, trocando-se os nomes das operações e eventos para nomes mais significativos e colocando as operações em locais mais apropriados. Por exemplo, no caso da oficina auto-elétrica e mecânica, a operação “lançamento_prim_parte” foi renomeada “abre_ordem_serv” e a operação “lançamento_seg_parte” foi renomeada “fecha_ordem_serv”. As operações que estavam nos grupos “Relatórios” e “Impressos” foram reunidas em um grupo chamado “Impressão”. As operações dos grupos “Manutenção_de_arquivos” e “Manutenção_de_tabelas” foram reunidas no grupo “Atualização_arquivos”.

```
lifecycle Auto-elétrico e Mecânica:=
  (Lançamento | Atualização_arquivos | Impressão | Consulta )*
Lançamento = ( abre_ordem_serv . [#Ordem_Serviço_impressa] ) | modifica_ord_serv |
  elimina_ord_serv | ( fecha_ordem_serv . [#Relat_serviços_executados] ) |
  insere_atendimento_balcão | altera_atendimento_balcão | elimina_atendimento_balcão |
  insere_compra | altera_compra | elimina_compra | insere_pedido | altera_pedido |
  elimina_pedido | insere_conta_a_pagar | insere_conta_a_receber |
  altera_conta_a_pagar | altera_conta_a_receber | elimina_conta_a_pagar |
  elimina_conta_a_receber | quita_conta_a_pagar | quita_conta_a_receber
Atualização_arquivos = ( insere_peca | insere_cliente | insere_fabricante | insere_fornecedor |
  insere_empregado | insere_tipo_veiculo | insere_valor_de_trabalho |
  insere_tempo_de_reparo | insere_componente | insere_mao_de_obra |
  altera_peca | altera_cliente | altera_fabricante | altera_fornecedor | altera_empregado |
  altera_tipo_veiculo | altera_valor_de_trabalho | altera_tempo_de_reparo |
  altera_componente | altera_mao_de_obra | elimina_peca | elimina_cliente |
  elimina_fabricante | elimina_fornecedor | elimina_empregado | elimina_tipo_veiculo |
  elimina_valor_de_trabalho | elimina_tempo_de_reparo | elimina_componente |
  elimina_mao_de_obra | atualiza_tabela_sugestao_de_precos )
Impressão = ( rel_ord_serv_pendentes | rel_ord_serv_do_mês | rel_pecas_estoque_critico |
  rel_pecas_em_estoque | rel_precos_por_fabricante | rel_equivalencia_entre_pecas |
  rel_pecas_por_componente | rel_pecas_adquiridas_fora | rel_pedidos_pendentes |
  rel_pedidos_pendentes_por_fornecedor | rel_compras_mês | rel_itens_cancelados |
  rel_vendas_mês | rel_contas_a_pagar | rel_contas_a_receber | rel_balanco_mensal |
  rel_cliente_ord_alfabet | rel_cliente_por_entidade | rel_fornecedor_ord_alfabet |
  rel_fornecedor_por_ramo | rel_fabricante | rel_empregado | rel_componentes |
  rel_veiculos | rel_mao_de_obra | rel_sugestao_precos | rel_servicos_eletricos |
  rel_servicos_mecanicos | cheques | duplicatas | ord_servico | termo_garantia |
  recibo | cancelamento_protestos | carta_cancelamento | vales |
  etiq_cod_barra_por_fornecedor | etiq_cod_barra_por_peca |
  etiq_cod_barra_por_compra | etiq_cod_barra_pecas_miudas | etiq_por_estantes |
  etiq_clientes | etiq_pessoa_juridica | etiq_pessoa_fisica | etiq_cli_por_entidade |
  etiq_clientes_desejados | etiq_clientes_duplicatas | etiq_fornecedores )
Consulta = ( cons_peca_letra_inicial | cons_peca_fabricante | cons_peca_codigo_barras |
  cons_peca_codigo_original | cons_equivalencia_peca | cons_estatistica_custos |
  cons_fornecedores | cons_vendas | cons_empregados | cons_compras |
  cons_componentes | cons_clientes )
```

Figura 17 – MCVS - Modelo de Ciclo de Vida do Sistema

2.4.3 – Modelo de Operações do Sistema

O Modelo de Operações do MASA foi refeito, de acordo com as novas classes do Modelo de Objetos, obtendo o Modelo de Operações do MAS (MOpS). A figura 18 mostra o esquema para a operação “abre_ordem_serv” do MOpS, correspondente à operação “lançamento_prim_parte” mostrada no capítulo anterior. O apêndice B mostra mais algumas operações importantes do sistema, complementando o Modelo de Operações do MAS.

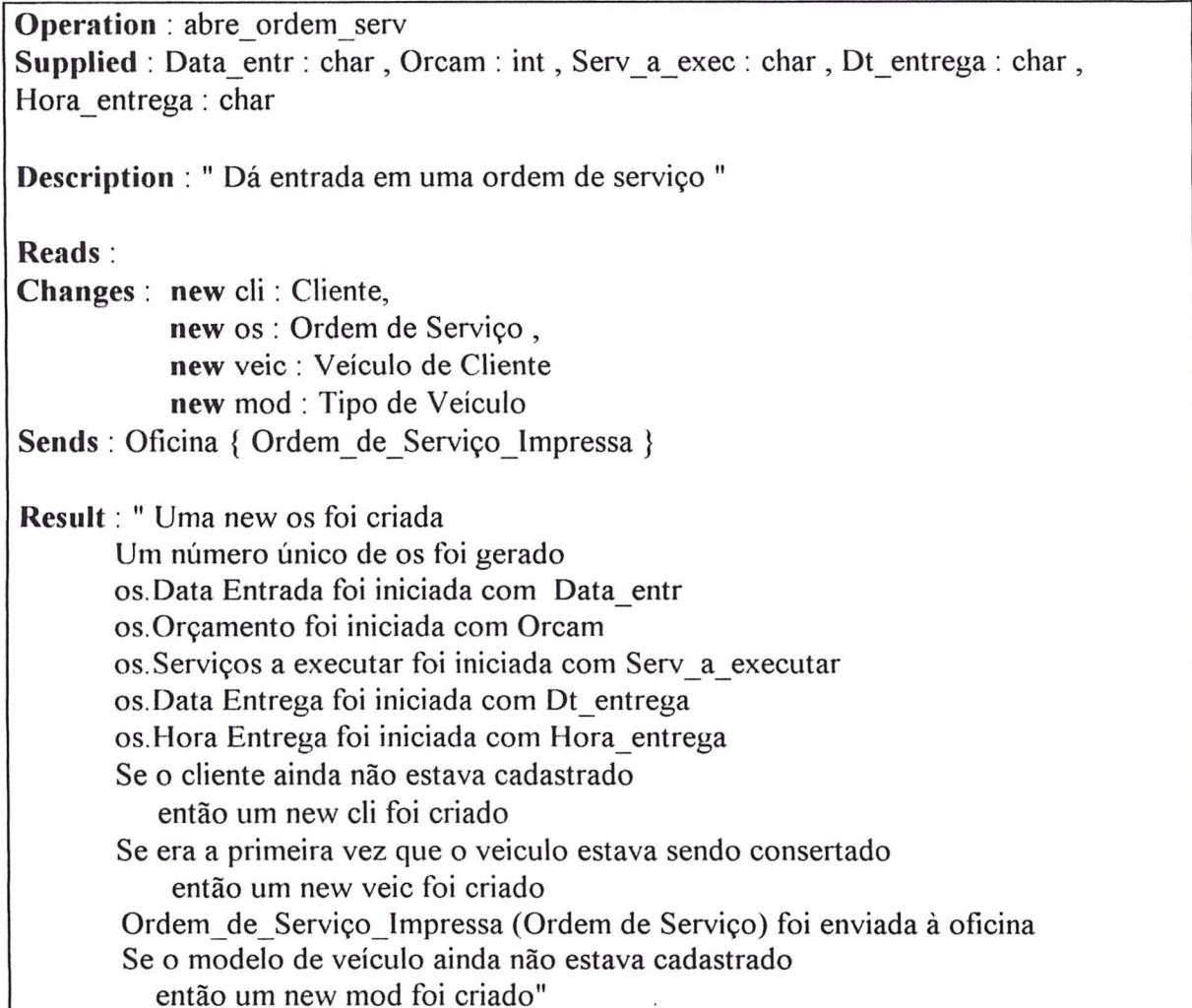


Figura 18 -- Esquema para a operação “abre_ordem_serv” do MOpS

2.5 – Passo 4: Correspondência entre o MAS e o MASA

Nesse passo faz-se o mapeamento do Modelo de Análise do Sistema (MAS) para o Modelo de Análise do Sistema Atual (MASA), descrevendo a relação entre as classes e métodos do MAS com os do MASA.

Graças ao nível de detalhamento com que foi feita a transição do MASA para o MAS, a realização deste passo foi facilitada. A tabela 1 mostra a correspondência entre o MAS e o MASA para a classe Cliente e seus atributos. A tabela 2 mostra a correspondência entre algumas operações do MAS e do MASA.

Vale observar que no mapeamento não foram especificados os métodos do MAS, como em [Pen 96], pois este trabalho possui outros objetivos mais específicos. Porém, está sendo desenvolvido um outro trabalho no qual os procedimentos do sistema legado são desdobrados em sub-procedimentos (ou “segmentos”), cada qual lidando com apenas uma classe. Numa futura reengenharia orientada a objetos, esses sub-procedimentos darão origem aos métodos do MAS. Poderá, então, ser feita a correspondência entre os métodos do MAS e os procedimentos do MASA.

Tabela 1 – Correspondência entre uma classe do MAS e o arquivo do MASA

MAS	MASA
Modelo de Objetos Classe: Veículo de Cliente	Arquivo Cliveic
Atributos	Campos do Arquivo
- - Placa Cidade Quilometragem Nro de meses Nro sequencial	codcliente codveiculo placa cidade hskm meses nroveiculo

Tabela 2 – Correspondência entre algumas operações do MAS e do MASA

MAS	MASA
Abre_Ordem_Serv	Lançamento_prim_parte
Fecha_Ordem_Serv	Lançamento_seg_parte
Inserere_Compra	Inserção_compra
Inserere_peça	Inserção_peças

3 - Conclusões

3.1 - Considerações Iniciais

O detalhamento efetuado definiu novas decisões a serem tomadas na transição do MASA para o MAS e acrescentou algumas diferenças de notação em relação ao trabalho de Penteadó [Pen 96]. Como exemplo de novas decisões, as classes, atributos e relacionamentos permaneceram sem nomes significativos até a etapa 6, quando foi adquirido um maior conhecimento sobre o sistema, possibilitando a atribuição de nomes compatíveis com o domínio de aplicação. Na passagem do MOS-1 para o MOS-2, relacionamentos originados de tabelas do MASA foram nomeados temporariamente com o nome da tabela original. Como exemplo de diferenças de notação, convencionou-se que as chaves das classes e as chaves estrangeiras teriam uma notação especial (fonte em negrito e sublinhada, respectivamente).

3.2 – Contribuições deste trabalho

Este trabalho possibilitou o detalhamento do processo de generalização do modelo de análise de um sistema legado para um modelo de análise orientado a objetos. Ao aplicar o Fusion/RE a um sistema comercial, foram documentadas todas as modificações feitas nos modelos durante a transição do MASA para o MAS, passando-se ao todo por sete etapas.

Na primeira etapa os pseudo-relacionamentos existentes no MOSA tornaram-se relacionamentos de fato, retirando-se os atributos do tipo “chave-estrangeira”. Na segunda etapa algumas tabelas do MOSA foram transformadas em relacionamentos do tipo “muitos para muitos”. Na terceira etapa foram identificadas as agregações e especializações de classes. Na quarta etapa foram identificadas algumas das entradas do sistema e os agentes externos que interagem com o mesmo. Na quinta etapa identificaram-se algumas das saídas do sistema e os agentes externos com elas envolvidos. Na sexta etapa foi feita a generalização de nomes referentes ao modelo produzido na quarta etapa e foram acrescentados ao modelo os demais atributos das classes. Na sétima etapa foi feita a generalização de nomes referentes ao modelo produzido na quinta etapa. Os modelos obtidos nas duas últimas etapas compõem o modelo de objetos do sistema final.

3.3 - Conclusões Finais

No exemplo adotado as sete etapas descritas foram suficientes para o entendimento do raciocínio utilizado na realização da generalização. Porém, em outros casos, outras etapas poderiam ser necessárias.

Por exemplo, na ausência de documentação do sistema legado, seria necessário detalhar melhor a etapa de revitalização da arquitetura do sistema, estudando a melhor maneira de extrair do código fonte as informações necessárias para tal. A existência de falhas de projeto da base de dados, por exemplo, implicaria na necessidade de inclusão de uma ou mais etapas para efetuar sua normalização. A utilização de uma linguagem de programação diferente, por exemplo uma linguagem de mais baixo nível, poderia aumentar o número de classes no MOSA que desapareceriam no MOS.

3.4 - Sugestões para novos trabalhos

Já estão em fase inicial trabalhos semelhantes com outros sistemas, como por exemplo um sistema de controle de pacientes de consultório médico, um sistema de controle ambulatorial e um sistema de malharia. Visa-se com isso verificar a adequação das etapas adotadas neste trabalho a outros análogos, procurando-se avaliar em que medida a ausência de documentação e a existência de falhas de projeto influenciam na definição dessas etapas.

Bibliografia

- [Bec89] Beck, K. e Cunningham, W. - A laboratory for teaching object-oriented thinking, ACM OOPSLA'89 Conference Proceedings, 1989.
- [Boo91] Booch, G. - Object-oriented Design with Applications, Benjamin Cummings, Redwood City, CA, 1991.
- [Bra90] Braga, Rosana T.V. - Manual de Uso do Sistema de Oficina Auto-elétrica e Mecânica, Documento de Trabalho, São Carlos, 1990.
- [Col94] Coleman D. et al, Object Oriented Development - The Fusion Method, Prentice Hall, 1994.
- [Oli91] Oliveira, Jacqueline A. e Braga, Rosana T.V. - Documentação Técnica do Sistema de Oficina Auto-elétrica e Mecânica, Documento de Trabalho, São Carlos, 1991.
- [Pen96] Penteado, R.D. Um método para Engenharia Reversa Orientada a Objetos, Tese de Doutorado, USP, Instituto de Física de São Carlos, Jan. 1996.
- [Pen96a] Penteado, R.D., Germano, F.; Masiero, P.C. - An Overall Process Based on Fusion to Reverse Engineer Legacy Code. III Working Conference on Reverse Engineering - IEEE. Monterey - California, pp. 179-188, 1996.
- [Pen96b] Penteado, R.D., Germano, F.; Masiero, P.C. - Melhoria na Manutenibilidade de Software Legado com Base em Engenharia Reversa. VII Conferência Internacional de Tecnologia de Software - Qualidade de Software, Curitiba - PR, pp. 155-169, 1996.
- [Rum91] Rumbaugh, J.; Blaha, W. Premerlani, Eddy, F. e Lorensen, W. - Object-Oriented Modeling and Design. Prentice Hall International, Englewood Cliffs, NJ, 1991.

Apêndice A – Esquemas para mais algumas operações do MOpSA

Operation : lançamento_seg_parte

Description : " Faz o lançamento da segunda parte da ordem de serviço "

Reads : **Supplied** : Nro_OS: int, Data_saida : char,
col peca_os (quant : int , tipo:int, codigo:int, precoun : int),
col serv_os (codigo:int, empr: int, preco: float)

e: Empregad,
p: Pecas

Changes : **new** col pos: Ospeca,
new col mos: Osmobra,
new cr : Contarec,
new op: Outpecas,
os: Ordserv

Sends : cliente: { Relat_serviços_executados }

Result : "os . datacompra foi iniciada com Data_saida
Para cada membro da coleção peca_os
Uma new pos foi criada
pos . osnro foi iniciada com Nro_OS,
pos . quantidade foi iniciada com Quant
Se tipo = 1 então
pos . codbarra foi iniciada com codigo
senão
Uma new op foi criada
op . codoutpeca foi iniciada com codigo
pos . codoutpeca foi iniciada com codigo
pos . precotot foi iniciado com precoun*quant
p. qtest foi decrementada de quant
Para cada membro da coleção serv_os
Uma new mos foi criada
mos . osnro foi iniciado com Nro_OS
mos . codmobra foi iniciado com codigo
mos . Codemp foi iniciado com empr
mos . preco foi iniciado com preco
Para i variando de 1 até Nro_parc
Uma new cr foi criada
cr . datavenc foi iniciada com data + 30
cr . parcela foi iniciada com i
cr . valor foi iniciada com valor total / Nro_parc"

Operation : inserção_compra

Description : " Faz a inserção de um compra de peças "

Reads : **Supplied :** Nro_pedido : int, Nome_fornec : char, Nota_fisc : char
col itens_comp (quant : int , precoun : int), Nro_parce: int

Changes : new col pc: Itemcomp
New col cp : Contapag,
New c : Compras,
p: pecas

Sends :

Result : "Uma new c foi criada
Um número único de compra (nro_comp) foi gerado
c . nrocompra foi iniciado com nro_comp
Se havia um pedido pendente com número Nro_pedido então
c . Situação do pedido foi alterado para atendido
c . controle foi iniciado com Nota_fisc
Para cada membro da coleção itens_comp
Uma new pc comprada foi criada
pc . nrocompra foi iniciado com nro_comp
pc . quantidade foi iniciada com quant
pc . precounit foi iniciado com precoun
p. qtest foi incrementada de quant
Para i variando de 1 até Nro_parce
Uma new cp foi criada
cp . datavenc foi iniciada com data + 30
cp . parcela foi iniciada com i
cp . valor foi iniciada com valor total / Nro_parce"

Apêndice B – Esquemas para mais algumas operações do MOpS

Operation : fecha_ordem_serv

Supplied : Data_saida : char, col peca_os (quant : int , codigo:int, precoun : int),
col serv_os (codigo:int, empr: int, preco: float)

Description : " Dá saída a uma Ordem de Serviço, cadastrando as peças utilizadas, os serviços executados e as contas a receber"

Reads : e: Empregado,
p: Peça

Changes : new col pos: Peça da Ordem Serviço,
new col mos: Mão-de-obra da Ordem Serviço,
new cr : Conta a receber,
os: Ordem de Serviço

Sends : cliente: { Relat_serviços_executados }

Result : "os . DataSaida foi iniciada com Data_saida

Para cada membro da coleção peca_os

Uma new pos foi criada

pos . Quantidade foi iniciada com quant

pos . Preço unitário foi iniciado com precoun

p. Quant. Em estoque foi decrementada de quant

Para cada membro da coleção serv_os

Uma new mos foi criada

mos . Preço com preco

Para i variando de 1 até Nro_parc

Uma new cr:Conta a receber foi criada

cp . Data Vencimento foi iniciada com data + 30

cp . Nro Parcela foi iniciada com i

cp . Valor foi iniciada com valor total / Nro_parc"

Operation : insere_compra

Supplied : Nro_pedido : int, Nome_fornec : char, Nota_fisc : char
col itens_comp (quant : int , precun : int), Nro_parce: int

Description : " Cadastra uma compra de peças efetuada junto ao fornecedor,
acrescentando-as ao estoque "

Reads : -

Changes : new col pc: Peça comprada,
New col cp : Conta a pagar,
New c : Compra,
p: Peça do estoque
ped: Pedido

Sends :

Result : "Uma new c:Compra foi criada

Se havia um pedido pendente com número Nro_pedido então
ped . Situação do pedido foi alterado para atendido

c . Nro da Nota Fiscal foi iniciado com Nota_fisc

Para cada membro da coleção itens_comp

Uma new pc:Peca comprada foi criada

pc . Quantidade foi iniciada com quant

pc . Preço unitário foi iniciado com precun

p. Quant. Em estoque foi incrementada de quant

Para i variando de 1 até Nro_parce

Uma new cp:Conta a pagar foi criada

cp . Data Vencimento foi iniciada com data + 30

cp . Nro Parcela foi iniciada com i

cp . Valor foi iniciada com valor total / Nro_parce"