

# PUC

Monografias em Ciência da Computação  
nº 14/92

## **Modelagem Conceitual de Bancos de Dados com Redes de Petri - Um Estudo de Caso**

Sidney Dias da Silva  
Bruno Maffeo

Departamento de Informática

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO  
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453  
RIO DE JANEIRO - BRASIL**

PUC RIO - DEPARTAMENTO DE INFORMÁTICA

Monografias em Ciência da Computação, Nº 14/92

Editor: Carlos J. P. Lucena

Maio, 1992

**Modelagem Conceitual de Banco de Dados  
com Redes de Petri - Um Estudo de Caso \***

Sidney Dias da Silva

Bruno Maffeo

\* Este trabalho foi patrocinado pela Secretaria de Ciência e Tecnologia da Presidência da República Federativa do Brasil.

## Resumo

O Modelo de Entidades e Relacionamentos tem sido utilizado como ferramenta para a modelagem conceitual das propriedades estáticas dos dados no processo de projeto de bancos de dados. O surgimento de novas aplicações para bancos de dados, com requisitos de modelagem dos aspectos dinâmicos acrescidos aos tradicionalmente considerados, tem motivado a pesquisa de modelos capazes de representar conceitos como entidades, seus atributos e relacionamentos e as regras e eventos que afetam ou pelos quais são afetados.

Este trabalho aborda a utilização de Redes de Petri na representação de regras e eventos a nível conceitual, como extensão ao Modelo de Entidades e Relacionamentos. Uma alternativa de representação é considerada através do uso de Redes de Petri Compactas contemplando as propriedades estáticas e as propriedades dinâmicas relevantes do universo de discurso.

Essa alternativa é apresentada através de um estudo de caso, demonstrando a geração de um modelo homogêneo para dados e funções, com formalismo que facilita o mapeamento dos conceitos para níveis mais baixos de representação.

*Palavras-chave:* Modelo de Entidades e Relacionamentos, Redes de Petri de Alto Nível, modelagem conceitual de bancos de dados, modelagem de aspectos estáticos e dinâmicos no projeto de bancos de dados.

## Abstract

The Entity-Relationship Model has been used as a tool for the conceptual modeling of static properties during the process of designing database. New applications of databases, which require the modeling of dynamic aspects in addition to the aspects traditionally considered, have motivated the research of models with the capability of representing concepts such as entities, their attributes and relationships and the rules and events which affect them or are affected by them.

This work evaluates the use of Petri Nets for the representation of rules and events at the conceptual level, as an extension to the Entity-Relationship Model. An alternative representation is considered which employs Compact Petri Nets to model both the static and dynamic properties which are relevant in the universe of discourse.

This alternative is presented in a case study format demonstrating the generation of a homogeneous model for data and functions, with a formalism which facilitates the mapping of the concepts onto the lower representation levels.

*Keywords:* Entity-Relationship Model, High Level Petri Nets, databases conceptual modeling, simultaneous modeling of static and dynamic aspects in the project of databases.

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>1</b>
<b>2. MODELAGEM DOS ASPECTOS ESTÁTICOS E DINÂMICOS</b>	<b>3</b>
2.1 - Redes de Petri Elementares	3
2.2 - Redes de Petri de Alto Nível	6
2.3 - Tradução de Diagramas E-R em Modelos de Redes de Petri Compactas	8
<b>3. MODELAGEM DOS ASPECTOS DINÂMICOS NO MODELO E-R</b>	<b>14</b>
3.1 - Regras e Eventos	14
3.2 - Definição de Regras	15
3.3 - Adição de Regras ao Modelo E-R	15
3.4 - Modelagem de Execução de Regras com Redes Lugar-Transição	15
<b>4. PROPOSTA: O USO DE REDES COMPACTAS PARA MODELAGEM DE REGRAS</b>	<b>18</b>
4.1 - A Linguagem de Anotação	18
4.2 - A Modelagem de Regras com Redes Compactas	19
<b>5. ESTUDO DE CASO</b>	<b>21</b>
5.1- Regras	22
5.2 - A Representação do Modelo E-R como Rede de Petri	23
5.3 - Definição Parcial do UD (Conjuntos Constantes)	33
5.4 - Definição da LA	35
5.5 - A Rede Compacta	37
<b>6. CONCLUSÃO</b>	<b>39</b>
<b>BIBLIOGRAFIA</b>	<b>40</b>

# 1. INTRODUÇÃO

A evolução tecnológica verificada no passado recente (última década) propiciou o surgimento de novas aplicações para sistemas de banco de dados. Sistemas de informação de escritório (OIS), sistemas de hipertexto e projeto de software ajudado por computador (CASE), por exemplo, necessitam de suporte para armazenamento de textos, gráficos e imagens [MELO88, ZDON90, BERT91]. Outras classes de sistemas - como os voltados para suporte ao trabalho cooperativo e para o controle de processos em tempo real - apresentam como um dos requisitos adicionais a capacidade do sistema de gerência do banco de dados (SGBD) de detectar e assumir a iniciativa das ações em face da ocorrência de eventos específicos.

Os SGBDs usualmente empregados devem ser estendidos com funcionalidades capazes de atender aos requisitos das novas aplicações. Além de possibilitarem o armazenamento, modificação e recuperação dos elementos passivos dos sistemas de informação, os novos SGBDs têm também de manipular vários dos seus componentes dinâmicos, estejam eles associados a estruturas de dados de um objeto (dinâmica intra-objetos) ou associados ao estado global de um banco de dados (dinâmica inter-objetos). Um SGBD capaz de monitorar o estado de um banco de dados e de executar um conjunto de ações predefinido para os eventos apropriados é um SGBD ativo [TANA90, TANA91].

A capacidade de monitoramento do banco de dados e de iniciativa por parte de um SGBD, em função de mudanças no estado do banco de dados gerenciado, são derivadas das descrições dos dados e das regras definidas. O projeto de um banco de dados deve contemplar, desde as primeiras etapas do ciclo de vida do sistema, a representação e a especificação detalhada dos aspectos estáticos e dos aspectos dinâmicos relevantes da realidade considerada.

O modelo conceitual é usualmente o primeiro dos modelos produzidos no processo de projeto de um banco de dados. Na construção de representações da realidade são utilizadas ferramentas para a produção de modelos conceituais, isto é, modelos com elevado poder de captura do significado dos conceitos percebidos. A ferramenta de uso mais difundido para essa finalidade é o Modelo de Entidades e Relacionamentos (MER) [CHEN76].

O MER é reconhecido como uma ferramenta de alto nível. Na sua forma básica ele é voltado para a representação das propriedades estáticas da realidade sob a forma de entidades, seus atributos e relacionamentos. Para aumentar seu poder de representação, tem recebido diversas extensões, buscando seu ajuste a classes de problemas específicas.

Em [TANA90, TANA91] é apresentada uma proposta para extensão do MER, que acrescenta a capacidade de modelagem de regras aplicáveis aos dados representados no modelo conceitual. Segundo essa proposta, o formalismo necessário para a semântica da execução das regras é dado por Redes de Petri do tipo Lugar-Transição, fornecendo as bases para a tradução do modelo conceitual no modelo lógico. O modelo conceitual obtido nessa abordagem é um modelo heterogêneo, já que utiliza diferentes ferramentas para a representação das propriedades estáticas e das propriedades dinâmicas.

O objetivo deste trabalho é a pesquisa do uso de Redes de Petri Compactas [HEUS89, HEUS90] na produção de representações conceituais homogêneas dos aspectos dinâmicos e estáticos de uma dada realidade. As seções seguintes estão organizadas da seguinte maneira:

- Na Seção 2, é feita uma revisão sucinta de Redes de Petri, com a abordagem detalhada do tipo de Rede que é utilizado em nossa proposta.
- A Seção 3 apresenta uma visão geral da proposta de Tanaka et al. [TANA90, TANA91], completando o cenário para o trabalho desenvolvido.
- Na Seção 4, é realizada a modelagem de regras com o tipo de Rede de Petri abordado na Seção 2.

- A Seção 5 contém um exemplo de aplicação da modelagem conceitual de dados e regras.
- Finalmente, nossas conclusões são apresentadas na Seção 6.

## 2. MODELAGEM DOS ASPECTOS ESTÁTICOS E DINÂMICOS

Heuser [HEUS89,HEUS90] apresenta uma proposta de modelagem das propriedades estáticas e das propriedades dinâmicas de sistemas de informação, utilizando para isso a abordagem E-R [CHEN76] e Redes de Petri.

De acordo com essa proposta, as redes de Petri, embora possam modelar tanto os aspectos estáticos quanto os dinâmicos, levam à produção de modelos excessivamente complexos, motivo pelo qual surgiu a idéia de se combinar redes de Petri com diagramas E-R, buscando limitar a complexidade dos modelos, tanto sua complexidade intrínseca quanto sua complexidade devida ao porte.

### 2.1 - Redes de Petri Elementares

Redes de Petri, quando usadas em modelagem de sistemas, mostram-se particularmente eficazes nas situações em que a realidade modelada apresenta alto grau de distribuição e concorrência. Seu campo de aplicação vem crescendo continuamente, atingindo desde áreas como a modelagem de componentes de software e hardware, sistemas distribuídos, linguagens de programação, controle de processos e protocolos de comunicação até a modelagem de sistemas sócio-técnicos em geral. Foi o surgimento das redes de alto nível no início da década de 80 que tornou possível sua utilização em áreas como sistemas de informação, bancos de dados, automação de escritórios, interação homem-máquina e inteligência artificial [RICH85, OBER86, REIS86, VOSS86a, VOSS86b].

As Redes de Petri podem ser aplicadas de acordo com duas abordagens diferentes, denominadas de abordagem *F* e abordagem *C/E* [HEUS90]. Segundo a abordagem *F*, o comportamento de um sistema pode ser explicado em termos de processos e cada processo pode ser decomposto em fenômenos de dois tipos básicos, denominados de estado e transição. Nessa abordagem, considera-se uma sucessão cronológica de ocorrências de fenômenos do tipo transição (finalizam e iniciam a existência de fenômenos do tipo estado), o que leva à produção de redes acíclicas de grandes dimensões, podendo mesmo resultar em modelos de dimensões infinitas

A abordagem *C/E* (Condição/Evento) é baseada em um procedimento de abstração que consiste na classificação dos fenômenos individuais em classes de fenômenos. Os fenômenos do tipo estado são agrupados em classes que recebem o nome de condição e os fenômenos do tipo transição em classes que recebem o nome de evento. Com a abordagem *C/E* são produzidos modelos de dimensões finitas, aplicáveis às diversas áreas.

Os conceitos básicos de Redes de Petri são apresentados a seguir.

#### 2.1.1 - Elementos de uma Rede de Petri

Redes de Petri Elementares são representadas graficamente por malhas compostas por elementos pertencentes a dois tipos:

- lugar, representado por um círculo e definindo uma marca que pode assumir dois estados: presente ou ausente;
- conexão entre lugares, representado por um retângulo ligado por ramos aos círculos que simbolizam os lugares envolvidos na conexão. Uma conexão define uma alteração que, se

habilitada, pode ocorrer; quando ocorre, em geral modifica o estado das marcas definidas pelos lugares envolvidos.

Um lugar pode participar de uma conexão de quatro maneiras diferentes, cada uma correspondendo a um ramo possuindo um tipo específico de rótulo:

- ramo alterador de entrada, representado por uma seta simples apontando para o retângulo; neste caso, o ramo indica que, ocorrendo a alteração definida pela conexão, haverá o desaparecimento (consumo) da marca existente no lugar conectado;
- ramo alterador de saída, representado por uma seta simples, com origem no retângulo; o ramo indica que, quando ocorre a alteração definida pela conexão, haverá a geração (surgimento) de uma marca no lugar por ele apontado;
- ramo restaurador de entrada, representado por uma seta dupla apontando para o retângulo; o ramo indica, neste caso, que a existência da marca no local de entrada é considerada apenas para efeito de habilitação da alteração, não havendo desaparecimento da marca após a ocorrência da alteração;
- ramo restaurador de saída, representado por uma seta dupla com origem no retângulo; neste caso, o ramo indica que a alteração definida pela conexão só estará habilitada se não houver marca presente no lugar por ele apontado e que sua ocorrência não muda o estado da marca definida por esse lugar, isto é, a marca permanece ausente.

### 2.1.2 - Funcionamento Básico de uma Rede de Petri Elementar

Um lugar de uma Rede de Petri Elementar é visto como um depósito de conteúdo variável, resumido graficamente pela presença ou ausência de uma ficha no lugar. A presença ou não da ficha no lugar é referida como marca presente ou marca ausente. A fig. 2.1, extraída de [HEUS90], apresenta uma Rede de Petri Elementar com a indicação da marcação de um lugar (Semáforo aberto).

Uma conexão de uma Rede de Petri Elementar define uma alteração que, quando ocorre, afeta as marcas dos lugares a que se liga diretamente através de ramos alteradores. O efeito sobre os lugares pode ser resumido pelo desaparecimento das marcas dos lugares de entrada e o coincidente aparecimento das marcas nos lugares de saída.

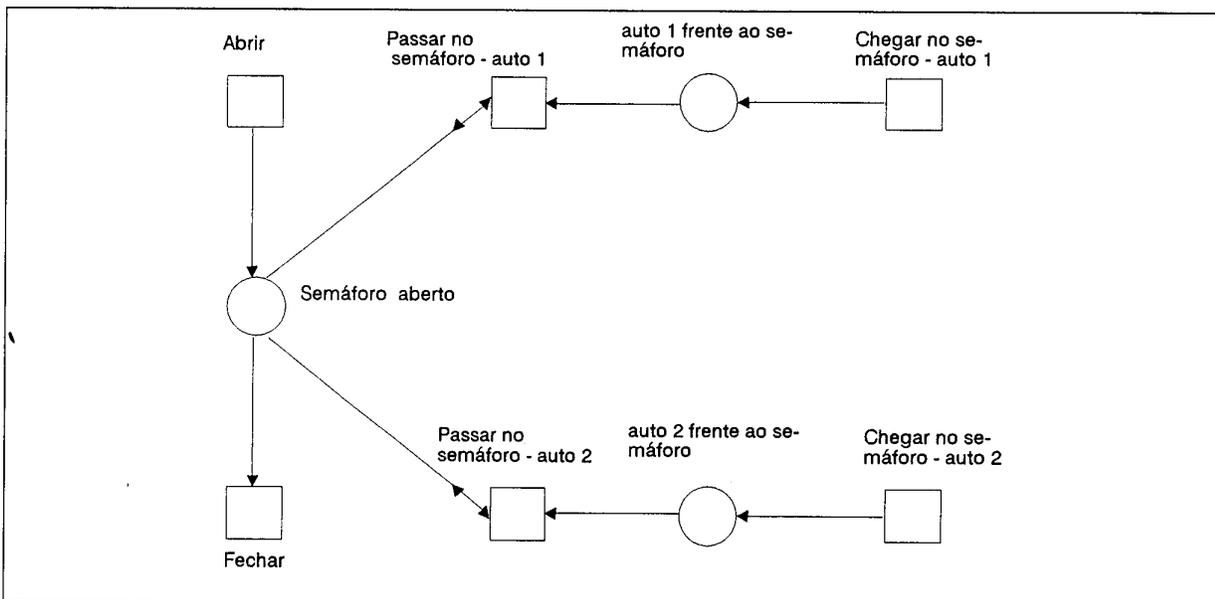


Fig. 2.1 - Exemplo de Rede de Petri Elementar [HEUS90]

Uma alteração pode ocorrer somente quando está habilitada. A habilitação exige o atendimento dos seguintes requisitos:

- as marcas de todos os lugares de entrada da conexão, caso os tenha, devem estar presentes;
- as marcas de todos os lugares de saída da conexão, caso os tenha, devem estar ausentes.

O ponto de partida do funcionamento da Rede é o estado global definido pela marcação inicial dos seus lugares. Uma alteração habilitada pode ocorrer a qualquer tempo, isto é, o fato de todas as exigências estarem satisfeitas não possibilita a determinação do momento da sua ocorrência, nem se a mesma irá efetivamente ocorrer.

### 2.1.3 - Modelagem de Alterações Concorrentes e de Alterações Conflitantes

A concorrência (independência) entre as alterações é refletida por alterações que não possuem marcas de entrada ou de saída comuns. Dessa forma, caso estejam habilitadas, a ocorrência em paralelo das alterações pode se verificar. A figura 2.2(a) mostra um exemplo de duas alterações concorrentes (independentes).

Uma situação de conflito, por outro lado, é refletida por alterações habilitadas com lugares de entrada ou de saída comuns. Sempre que uma situação dessas se verificar, a Rede estará indicando que apenas uma das alterações poderá ocorrer, embora não se possa dizer qual, nem prever-se quando (v. fig. 2.2.(b)).

### 2.1.4 - Semântica Associada a uma Rede de Petri Elementar

Uma Rede de Petri Elementar por si só não possui significado algum; para fins de modelagem de sistemas, uma interpretação deve ser associada a ela, visando atribuir significado a cada elemento no tipo de aplicação considerado.

A interpretação dada à presença ou ausência de uma marca em um lugar é o vigorar ou

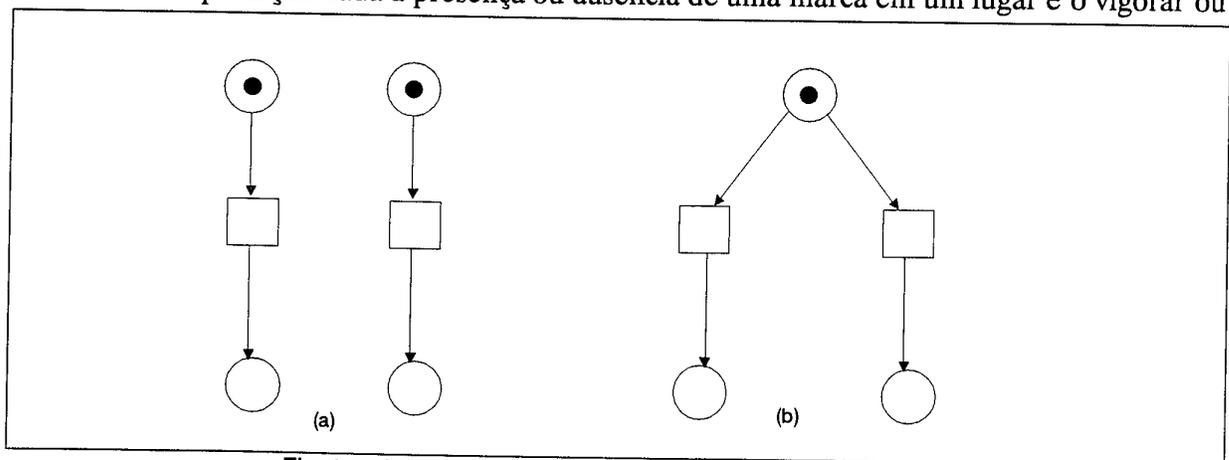


Fig. 2.2 - Independência e Conflito entre Alterações[HEUS90]

não de uma determinada condição; cada condição é vista como um estado local do sistema, independente dos demais estados locais considerados. Um estado (global) do sistema é definido pelo conjunto dos seus estados locais.

A interpretação que se dá, em termos de sistema, à ocorrência de uma alteração, é a ocorrência de um evento relevante para a modelagem do comportamento do sistema.

Considera-se, então, que a modelagem conceitual de sistemas de informação tem por objetivo representar, em nível elevado de abstração, propriedades estáticas e dinâmicas do sistema a ser modelado, através da definição de um conjunto de estados e um conjunto de transições entre esses estados. Um estado, a nível de sistema, é definido pelo vigorar - ou não -

de cada membro do conjunto de condições do sistema e as transições entre os estados são definidas em termos de conjuntos de eventos; cada evento, modelado por alteração definida por uma conexão da rede, quando ocorre, modifica o (não) vigorar das condições modeladas por marcas definidas pelos lugares ligados, através de ramos, à conexão considerada. Resulta daí o nome de redes para Modelagem Condição/Evento (CEM - Condition /Event Modeling).

## 2.2 - Redes de Petri de Alto Nível

Redes de Petri Elementares são utilizadas para representar dinâmica envolvendo instâncias de tipos de entidades. Quando é necessário tratar diversas instâncias, deve-se criar redes elementares cujo porte depende do número de instâncias envolvidas.

A dificuldade de modelar situações que envolvem um grande número de instâncias levou à criação de representações de mais alto nível de abstração, com poder de representação igual ao de redes elementares de porte muito maior.

A Rede de Petri utilizada no trabalho de Heuser é uma rede de alto nível, denominada Rede Compacta, considerada mais adequada à modelagem de sistemas de informação do que as redes elementares. Esse tipo de rede resultou da combinação de dois tipos de redes de alto nível (Rede Predicado/Transição e Redes Coloridas). A denominação "Rede Compacta" decorre do fato de serem produzidas, para um determinado sistema, redes de dimensões menores do que as da rede elementar equivalente.

A Rede Compacta proposta é um tipo particular de Rede de Petri, anotada com uma linguagem de lógica de predicados de primeira ordem e dotada de regras exatas de funcionamento.

A Rede Condição/Evento Compacta consta de :

- definição de linguagem da lógica de predicados de primeira ordem, através da especificação de seus símbolos não lógicos;
- Rede de Petri anotada com essa linguagem, da seguinte maneira:
  - a cada conexão é associada uma fórmula da linguagem, denominada fórmula da conexão; a fórmula é anotada dentro ou próximo do retângulo associado à conexão. Quando a fórmula é sempre avaliada por VERDADEIRO, pode ser omitida;
  - a cada ramo de conexão é associado um termo da linguagem, denominado termo do ramo; esse termo designa sempre um conjunto de elementos do Universo de Discurso (UD) pertencente à realidade modelada, subconjunto do domínio do lugar ao qual está ligado;
  - a cada lugar são associados dois termos constantes da linguagem - termos sem variáveis livres - que designam conjuntos do UD. Um dos termos, escrito entre parênteses, define o domínio do lugar, e é anotado junto ao símbolo gráfico, após o nome do lugar; o outro termo designa a marcação inicial do lugar e aparece na representação gráfica dentro do símbolo correspondente. Quando o termo que designa a marcação inicial é o conjunto vazio, a anotação não é efetuada;
- definição do UD, através de termos da linguagem de anotação que designam os elementos do Universo.

A figura 2.3 apresenta um exemplo de Rede Compacta.

### 2.2.1 - Significados Associados aos Lugares

O domínio de um lugar indica que elementos do UD podem estar presentes no lugar

considerado. A presença de um elemento no lugar é referenciada através de uma marca definida pelo par  $(l,e)$ , onde  $l$  identifica o lugar e  $e$  designa um elemento pertencente ao domínio do lugar.

Dessa maneira, podem ser construídas frases como "a marca  $(l,e)$  está presente (ausente)", significando que o elemento  $e$  está presente (ausente) no lugar  $l$ .

Quando o domínio de um lugar é omitido, significa que esse lugar define uma única marca (lugar elementar).

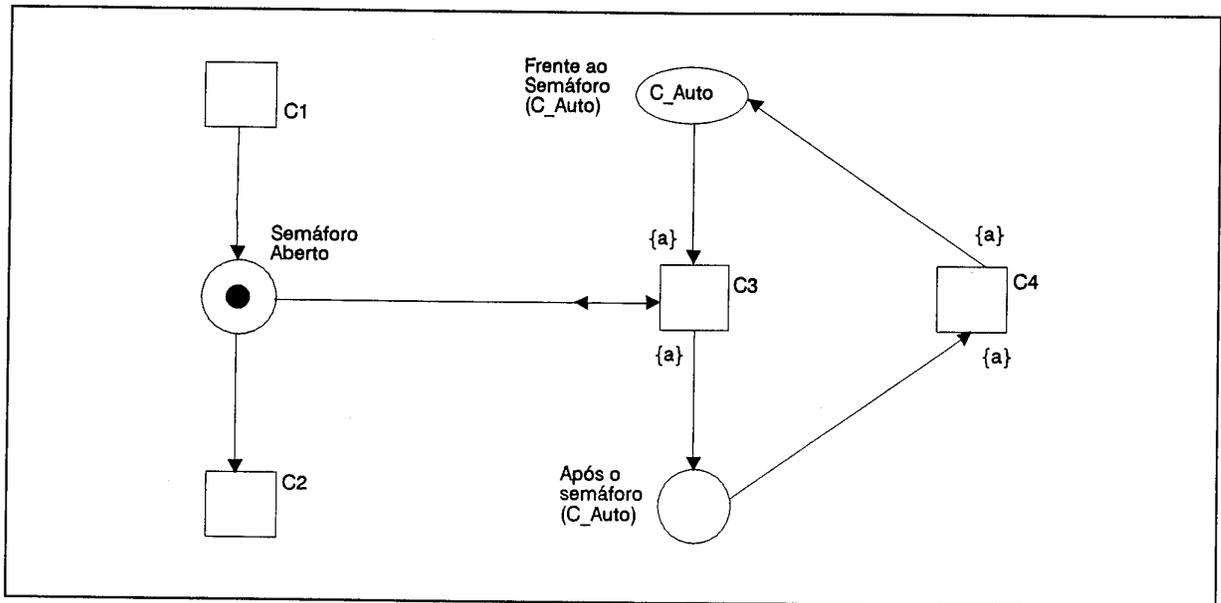


Fig. 2.3 - Exemplo de Rede Compacta [HEUS90]

### 2.2.2 - Significados Associados às Conexões

Os ramos das conexões são geralmente anotados com termos da linguagem de anotação, designando sempre conjuntos de entidades do UD, podendo haver casos em que o ramo é anotado com uma fórmula da linguagem de anotação. São casos excepcionais destinados a prover uma representação abreviada.

Uma conexão define uma alteração quando, para cada valoração de suas variáveis livres, são obedecidas as seguintes exigências:

- o conjunto de entidades designado pelo termo de cada ramo, sob a valoração considerada, deve fazer parte do domínio do lugar ao qual está ligado;
- a fórmula da conexão, quando presente, deve resultar, sob a valoração considerada, em verdadeira.

### 2.2.3 - Regras de Funcionamento da Rede

A rede CEM é apresentada como um modelo para representar relações entre pré-requisitos e consequências; os eventos são modelados com explicitação dos pré-requisitos e das consequências. Toda a dinâmica do sistema (transições entre estados) é expressa através das alterações que modelam as ocorrências de eventos.

O efeito da ocorrência de uma alteração é o desaparecimento, nos lugares de entrada, das marcas indicadas pelos termos dos ramos alteradores de entrada da conexão e o aparecimento coincidente, nos lugares de saída, das marcas indicadas pelos termos dos ramos alteradores de saída da conexão.

As demais regras apresentadas para Redes de Petri Elementares - habilitação de uma alteração, ponto de partida do funcionamento da rede, seqüência, concorrência e conflito de alterações e não determinação do momento da ocorrência de uma alteração - são válidas para as Redes Compactas.

## 2.2.4 - Modelagem de Propriedades Estáticas

Heuser [HEUS89,HEUS90] propõe a modelagem de propriedades estáticas de um sistema - como restrições de integridade, por exemplo - através de alterações que nunca estarão habilitadas. Dessa forma, através da afirmação de que as combinações de marcas presentes e ausentes jamais ocorrerão para uma dada alteração, é possível explicitar propriedades sobre conjuntos de dados.

As alterações que jamais estarão habilitadas são definidas por conexões mortas, sendo representadas por um retângulo com um *F* estilizado inscrito. Na figura 2.4 a conexão morta M1 expressa o fato de que uma dada pessoa não pode estar livre e ocupada ao mesmo tempo.

As conexões mortas são também utilizadas para restringir o conjunto de marcações alcançáveis. Uma marcação somente é alcançável pela ocorrência de alterações quando ela não é sucessora de marcação que habilite alguma conexão morta.

## 2.3 - Tradução de Diagramas E-R em Modelos de Redes de Petri Compactas

O Modelo E-R empregado por Heuser [HEUS89,HEUS90] é o Modelo apresentado por Chen [CHEN76], com as extensões adotadas por Setzer [SETZ86]. A partir do modelo estendido, são definidas representações equivalentes em Redes de Petri, conforme abordado a seguir.

### 2.3.1 - Tipos de Entidade

Um tipo de entidade do Diagrama E-R é traduzido por um lugar na Rede de Petri; o domínio desse lugar é o conjunto de todas as entidades do Universo de Discurso que compartilham o conjunto de atributos que caracteriza o tipo de entidade considerado. A figura 2.5 ilustra uma tradução do tipo AUTOR, sem considerar os tipos de atributo que o caracterizam.

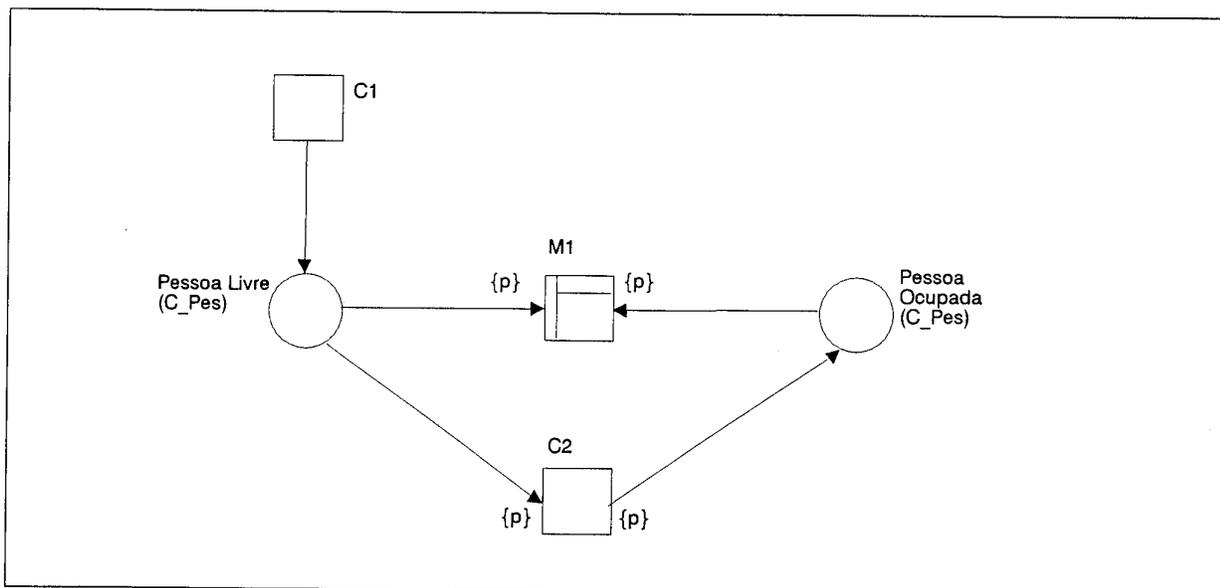


Fig. 2.4 - Exemplo de Rede com Conexão Morta [HEUS90]

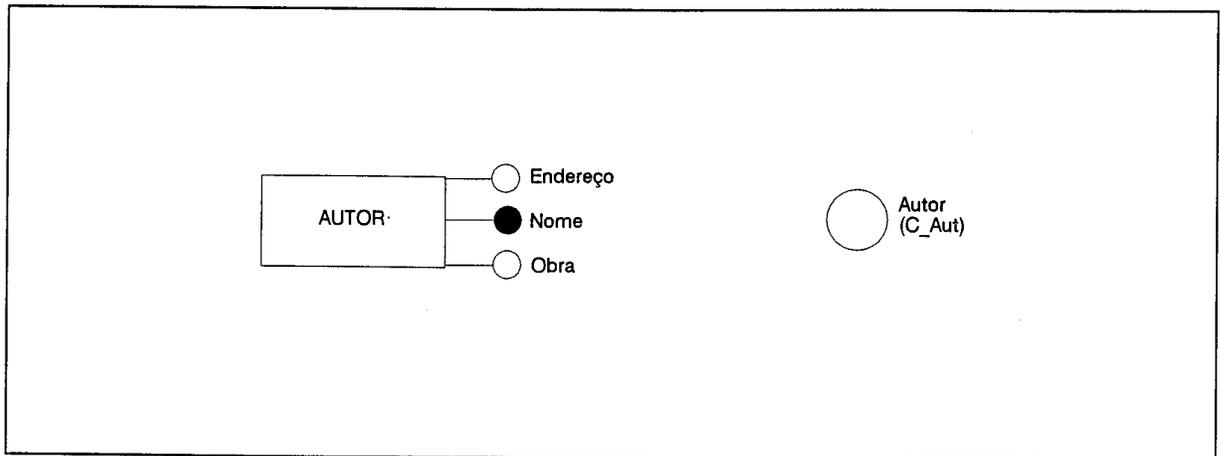


Fig. 2.5 - Tradução para Redes: Conjunto de Entidades [HEUS90]

### 2.3.2 - Tipos de Relacionamento

Um tipo de relacionamento é traduzido por um lugar, cujo domínio é o produto cartesiano dos tipos de entidade que a associação modela. Esse lugar está ligado aos lugares representativos dos tipos de entidades relacionadas através de conexões mortas, expressando que, para cada relacionamento presente, as entidades também estão presentes. A figura 2.6 apresenta um exemplo de tradução de tipo de relacionamento em rede de Petri.

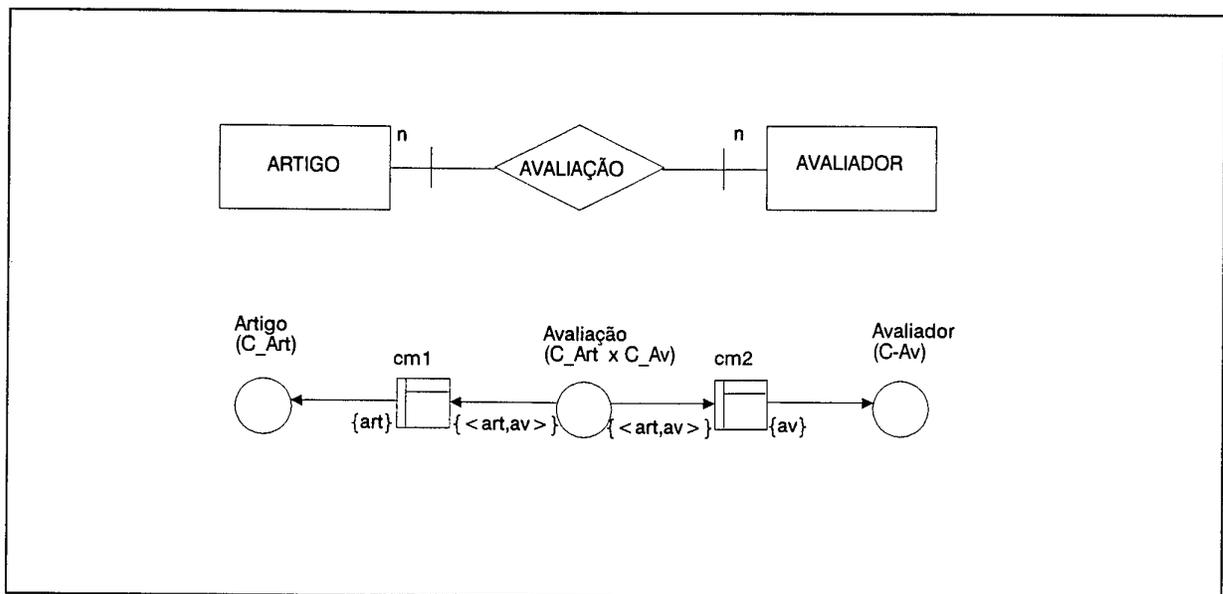


Figura 2.6

Tradução para Redes: Conjunto de Relacionamentos  $n:n$ , parcial [HEUS90]

A Rede de Petri assim construída modela relacionamentos da classe  $n:n$  e de natureza parcial. Para as demais classes e naturezas são necessárias conexões mortas adicionais na rede de Petri, da seguinte forma:

- A especificação de relacionamentos do tipo  $1:n$  exige a inclusão de uma conexão morta adicional sobre o lugar representativo do conjunto de relacionamentos, expressando que não existem duas tuplas com valores de atributos iguais e correspondentes ao tipo de entidade do lado  $1$  do relacionamento. A figura 2.7 ilustra uma situação do tipo.
- A especificação de relacionamentos com participação total exige a inclusão de uma conexão morta que explicita a existência de uma entidade nos lugares correspondentes às entidades para cada entidade presente no lugar correspondente ao relacionamento. A

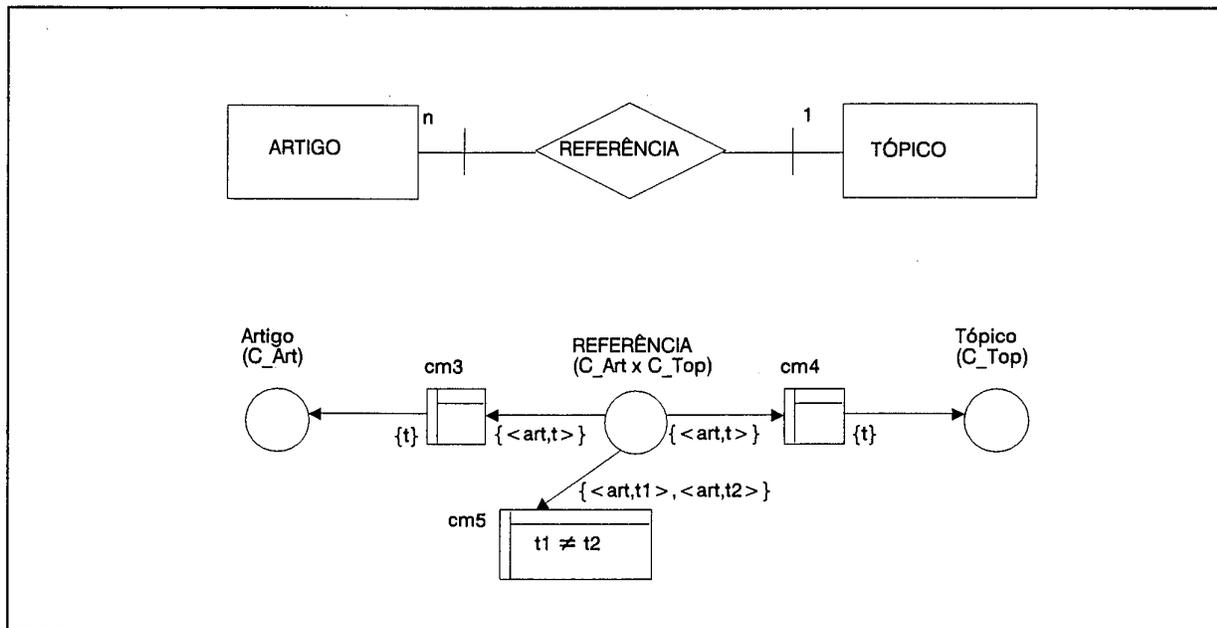


Figura 2.7  
Tradução para Redes: Conjunto de relacionamentos 1:n, parcial [HEUS90]

figura 2.8 apresenta um exemplo de relacionamento  $n:n$ , com participação total de ambas as entidades.

### 2.3.3 - Tipos de atributo

Um tipo de atributo do Diagrama E-R é traduzido pelos seguintes elementos da Rede de Petri:

- na linguagem de anotação da rede: um símbolo de constante designando o domínio do tipo de atributo. Definido como um conjunto de valores possíveis do atributo, esse

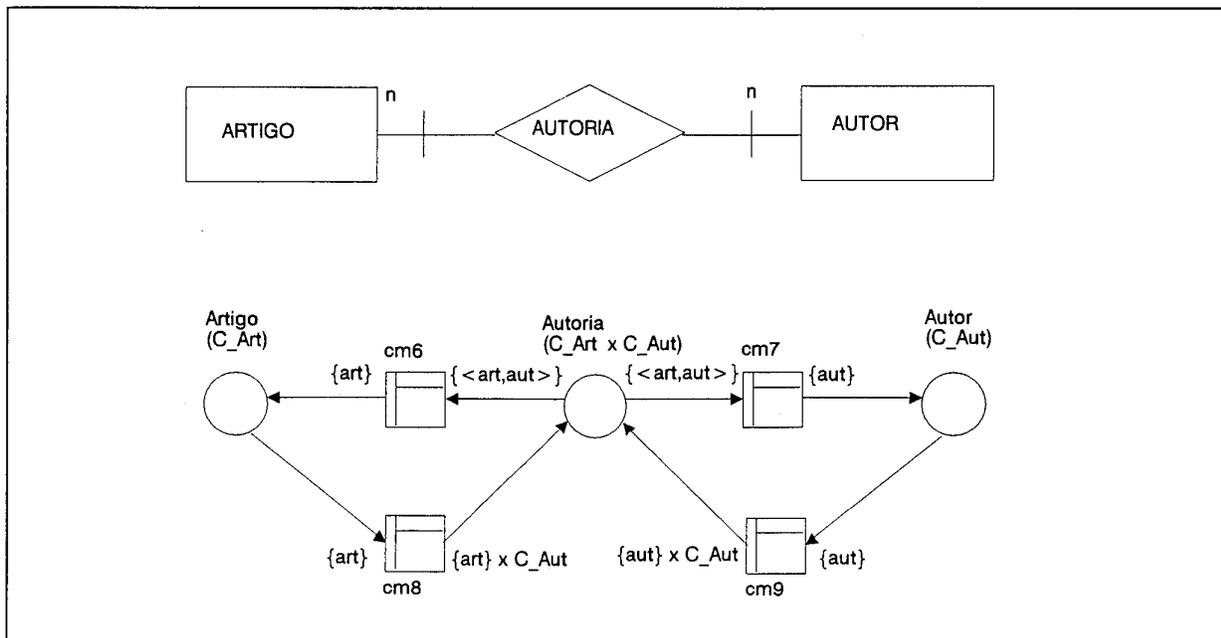


Figura 2.8  
Tradução para Redes: Conjunto de Relacionamentos  $n:n$ , total [HEUS90]

conjunto não sofre alterações, sendo anotado na rede e não modelado como um lugar exclusivamente designativo do tipo de atributo considerado;

- na rede:
  - um lugar, tendo como domínio o produto cartesiano do domínio do lugar correspondente ao tipo de entidade ou relacionamento ao qual está associado o atributo pelo conjunto de valores do atributo;
  - um par de conexões mortas; as conexões expressam a associação entre entidade e atributo, representando a restrição de integridade de que toda entidade (ou relacionamento, quando este possui atributos) tem associado a ela pelo menos um valor de atributo;
  - uma conexão morta expressando que, no lugar correspondente ao atributo, há somente uma tupla para cada entidade ou relacionamento com atributos;
  - para atributos identificadores, deve ser incluída uma conexão morta para garantir que, para cada valor de atributo, há somente uma tupla  $\langle entidade, valor\_do\_atributo \rangle$  no lugar correspondente ao atributo.

A figura 2.9 mostra um exemplo de tradução de atributos em Rede de Petri. Para simplificar a representação gráfica foi omitido o atributo obra.

A derivação de uma rede de Petri modelando os aspectos estáticos e dinâmicos de um sistema pode ser feita, segundo proposto por Heuser, em passos sucessivos, sumarizados a seguir:

- Modelagem dos aspectos dinâmicos: elaboração de um Diagrama de Fluxo de Dados-DFD, derivação de uma Rede de Petri do tipo Canal/Agência, derivação das redes compactas correspondentes a cada uma das agências e integração das redes compactas, para obtenção da rede global (este último, pelo observado, seria um passo opcional).
- Modelagem dos aspectos estáticos: elaboração do modelo conceitual com a abordagem

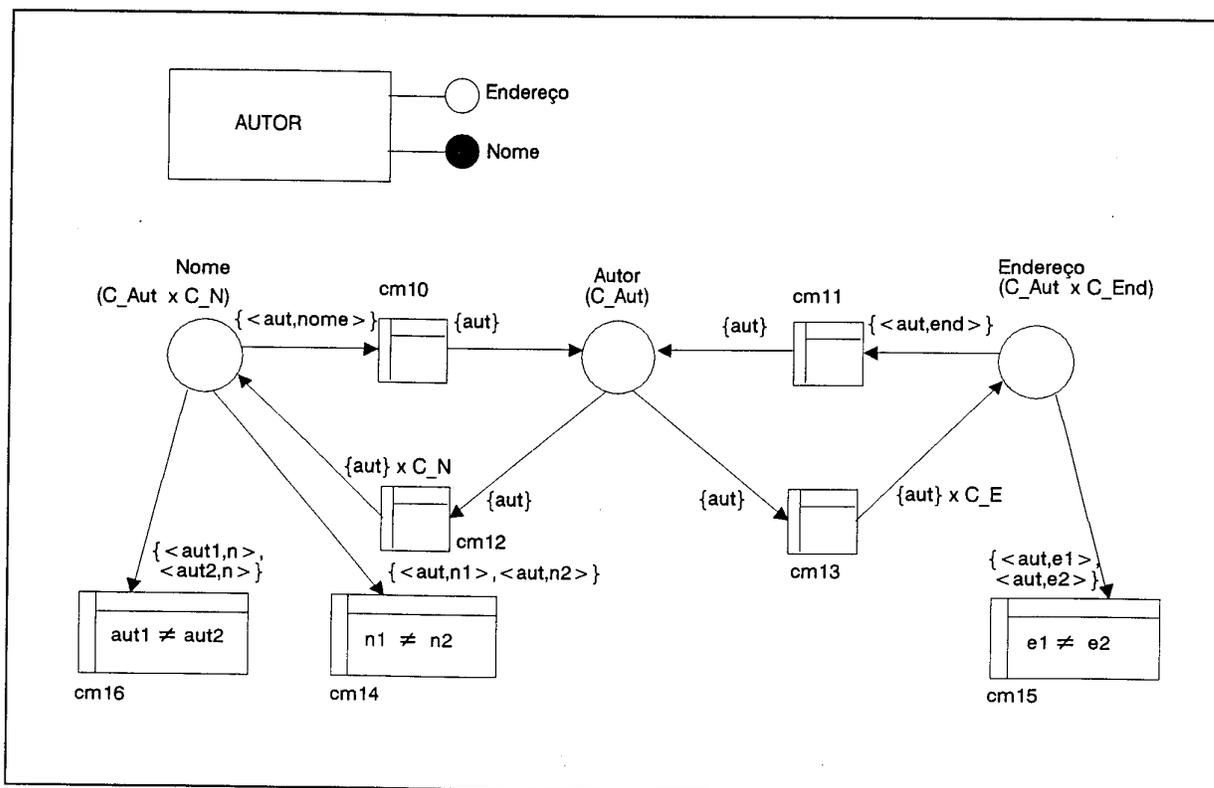


Figura 2.9  
Tradução para Redes: Conjunto de atributos [HEUS90]

E-R, derivação da rede compacta correspondente e inclusão das conexões mortas que modelam as demais restrições de integridade não expressas no Modelo E-R.

A rede final do sistema seria obtida pela integração de ambas as redes, oferecendo um modelo único das propriedades estáticas e dinâmicas relevantes para a modelagem da realidade sob exame.

### 2.3.4 - Hierarquias de Abstração: Generalização e Especialização

A generalização/especialização expressa no Diagrama E-R é traduzida para Rede de Petri da seguinte forma:

- cada conjunto de entidades de generalização e de especialização é traduzido por um lugar na Rede, e o seu domínio é definido conforme apresentado anteriormente no item 2.3.1;
- para cada lugar representativo de conjunto de entidades especializado é criada uma conexão morta, expressando que toda entidade ali presente também está presente no lugar representativo do conjunto de entidades genérico;
- uma conexão morta que expressa que os lugares representativos dos conjuntos especializados são disjuntos, isto é, que uma entidade só pode estar presente em um desses lugares.

A figura 2.10 apresenta um exemplo de tradução de hierarquia de abstração do tipo generalização/especialização para Rede de Petri. No caso de hierarquias que representem generalização total, há necessidade de uma conexão morta adicional, expressando que toda entidade presente no lugar representativo do conjunto de entidade genérica também está presente em exatamente um lugar representativo de conjunto de entidade especializada (v. figura 2.11).

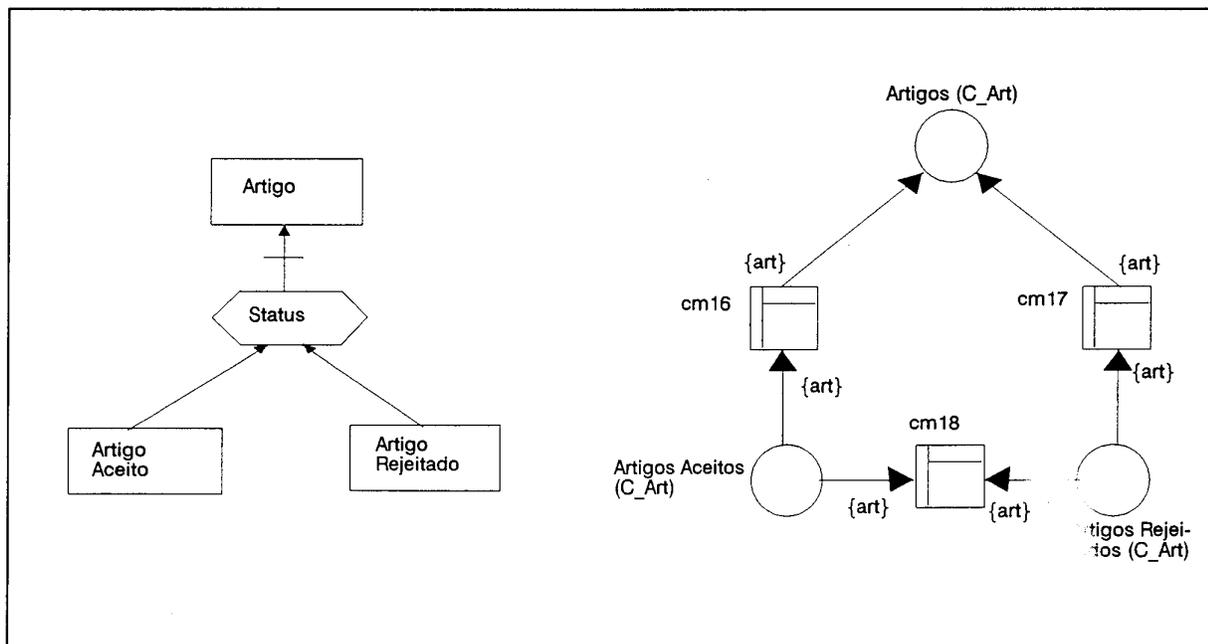


Figura 2.10  
Tradução para Redes: Generalização Parcial [HEUS90]

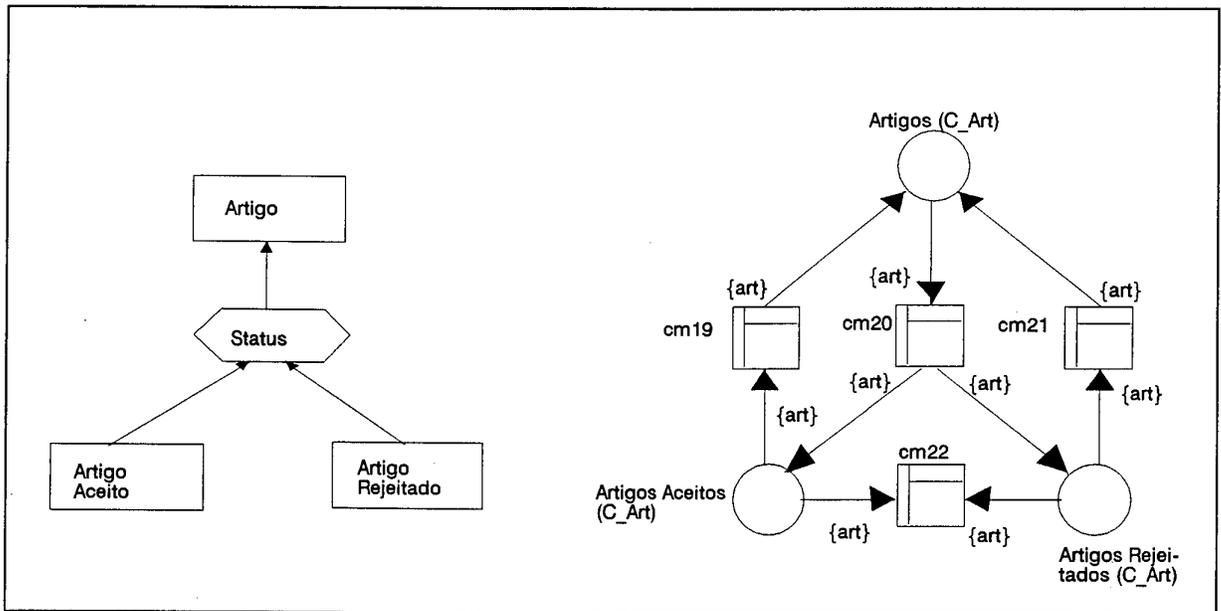


Figura 2.11  
Tradução para Redes: Generalização Total [HEUS90]

### 3. MODELAGEM DOS ASPECTOS DINÂMICOS NO MODELO E-R

Tanaka et al. [TANA90, TANA91] apresentam uma proposta de extensão do Modelo E-R, com vistas à modelagem conceitual de regras associadas a estados dos objetos de dados modelados e seus valores. Dessa maneira, um novo tipo de objeto passa a ser modelado, representando essas regras em termos de situação-ação. No mapeamento dos conceitos para o nível lógico, Tanaka propõe que o refinamento da especificação de regras seja feito com a utilização de Redes de Petri do tipo Lugar/Transição como modelo de execução.

Tanaka considera que o acréscimo de regras aos diagramas de Entidades e Relacionamentos resulta em um instrumento no nível conceitual que se revela bastante útil no processo de projeto de bancos de dados.

#### 3.1 - Regras e Eventos

Uma situação é definida como sendo uma dupla  $\langle e, c \rangle$ , onde  $e$  é um evento e  $c$  a condição associada.

Um evento acontece em algum instante no tempo e, teoricamente, não tem duração; é feita a distinção entre a ocorrência do evento e a ação que ele causa, executada por algum agente.

Uma outra distinção é feita, entre classes de eventos e instâncias de eventos, por um lado, e classes de regras e instâncias de regras, por outro, de forma análoga à formulada por Chen [CHEN76] para os objetos conceituais entidades e relacionamentos que possuem classes - denominadas de conjuntos de entidades ou de relacionamentos - e instâncias.

Como exemplo, considere-se a classe de eventos "empregado eliminado" e uma instância dessa classe, "todos os empregados que trabalham no projeto ALPHA foram eliminados".

Os eventos podem sinalizar simplesmente que alguma coisa ocorreu - como "motor parado", "log cheio" - ou transmitir informação na forma de atributos de evento, similares aos atributos de entidades ou relacionamentos.

Os eventos são agrupados em dois tipos básicos: os que ocorrem em dados armazenados no banco de dados e os que são externos ao conteúdo do banco de dados.

Um evento de banco de dados está associado a uma operação de consulta ou de modificação (inserção, eliminação ou alteração) do seu conteúdo - entidades, atributos ou relacionamentos.

Um evento externo é visto como sinais percebidos no domínio da aplicação tais como a paralisação de um motor, o enchimento de um buffer ou a inatividade de uma linha de comunicação.

Um tipo especial de evento é também distinguido, denominado evento temporal, associado a condições que se tornam verdadeiras em função da passagem do tempo.

A sintaxe empregada para a definição de eventos é a seguinte:

```
<evento> ::= <evento_BD> | <evento_externo> | <evento_temporal>  
<evento_BD> ::= <objeto_de_dados> modificado | <atributo> modificado  
| <objeto_de_dados> consultado | <atributo> consultado
```

Na definição,  $\langle \text{objeto\_de\_dados} \rangle$  é considerado como sendo um conjunto de entidades ou de relacionamentos, e  $\langle \text{atributo} \rangle$  um atributo de um  $\langle \text{objeto\_de\_dados} \rangle$ .

### 3.2 - Definição de Regras

Uma regra é formada por *nome\_da\_regra*, *evento\_disparador* e um conjunto não-vazio de triplas  $\langle \text{condição}, \text{ação}, \text{eventos\_resultantes} \rangle$ ; como interpretação informal, é apresentado que, quando um *evento\_disparador* é detectado, a condição é avaliada e, se verdadeira, a ação correspondente é executada, produzindo os *eventos\_resultantes* associados.

O *evento\_disparador* é visto como uma combinação de eventos de entrada que, uma vez detectada, dispara a regra. É utilizada a sintaxe:

$\langle \text{evento\_disparador} \rangle ::= \langle \text{evento} \rangle \mid \text{evento OR } \langle \text{evento\_disparador} \rangle$

Em cada tupla  $\langle \text{condição}, \text{ação}, \text{eventos\_resultantes} \rangle$  a condição deve ser avaliada, e a ação correspondente só é executada quando a condição se revela verdadeira; cada condição é apresentada sob a forma de predicado, resultante de uma consulta ao banco de dados. Uma regra pode ter condição *nenhuma*, isto é, a ação será sempre executada quando o evento disparador for detectado.

Uma ação é uma sequência de comandos traduzidos em termos de operações sobre os dados do banco de dados ou de operações externas que não afetam ou consideram o conteúdo do banco de dados. Um tipo especial de ação,  $\langle \text{rejeitar\_tentativa\_de\_operação} \rangle$ , é considerada para especificar regras que assegurem a proibição de determinadas operações de modificação. A sintaxe proposta para ação é:

$\langle \text{ação} \rangle ::= \langle \text{ação\_básica} \rangle \mid \langle \text{ação\_básica} \rangle ; \langle \text{ação} \rangle$

$\langle \text{ação\_básica} \rangle ::= \langle \text{operação\_de\_dados} \rangle \mid \langle \text{processo\_externo} \rangle$

$\langle \text{operação\_de\_dados} \rangle ::= \langle \text{operação\_de\_modificação} \rangle \mid \langle \text{operação\_de\_consulta} \rangle$

$\langle \text{operação\_de\_modificação} \rangle ::= \langle \text{operação\_de\_inserção} \rangle \mid \langle \text{operação\_de\_eliminação} \rangle$   
 $\mid \langle \text{operação\_de\_alteração} \rangle$

*Eventos\_resultantes* é o conjunto de eventos de saída que são originados pela execução de regras. O conjunto pode ser definido como *nenhum*, o que significa que nenhum evento relevante resulta da execução da ação.

### 3.3 - Adição de Regras ao Modelo E-R

No Diagrama E-R, uma regra é modelada por um paralelograma, com arcos dirigidos mostrando os eventos de entrada e saída, e possíveis dados adicionais. No caso de eventos de BD, os arcos conectam a regra aos objetos de dados ou atributos que servem para especificar o evento, com a anotação do tipo de evento (inserção, modificação etc.). A figura 3.1 mostra um diagrama E-R acrescido de regras.

As regras podem capturar uma variedade de características, desde políticas organizacionais de alto nível até restrições de integridade.

### 3.4 - Modelagem de Execução de Regras com Redes Lugar-Transição

A modelagem estendida com a incorporação de regras é adequada ao nível conceitual de abstração do processo de projeto de banco de dados; sua utilização em níveis mais baixos de abstração implica um processo de refinamento que exige a tradução dos conceitos em construções próprias dos níveis lógico e físico.

O refinamento proposto consiste na tradução da execução de regras para o modelo de execução através do uso de uma abordagem baseada em Redes de Petri do tipo Lugar-Transição.

No modelo adotado, existem dois tipos de lugares representando eventos e condições, e dois tipos de barras de transição, representando detecção de eventos e ações. A diferenciação é

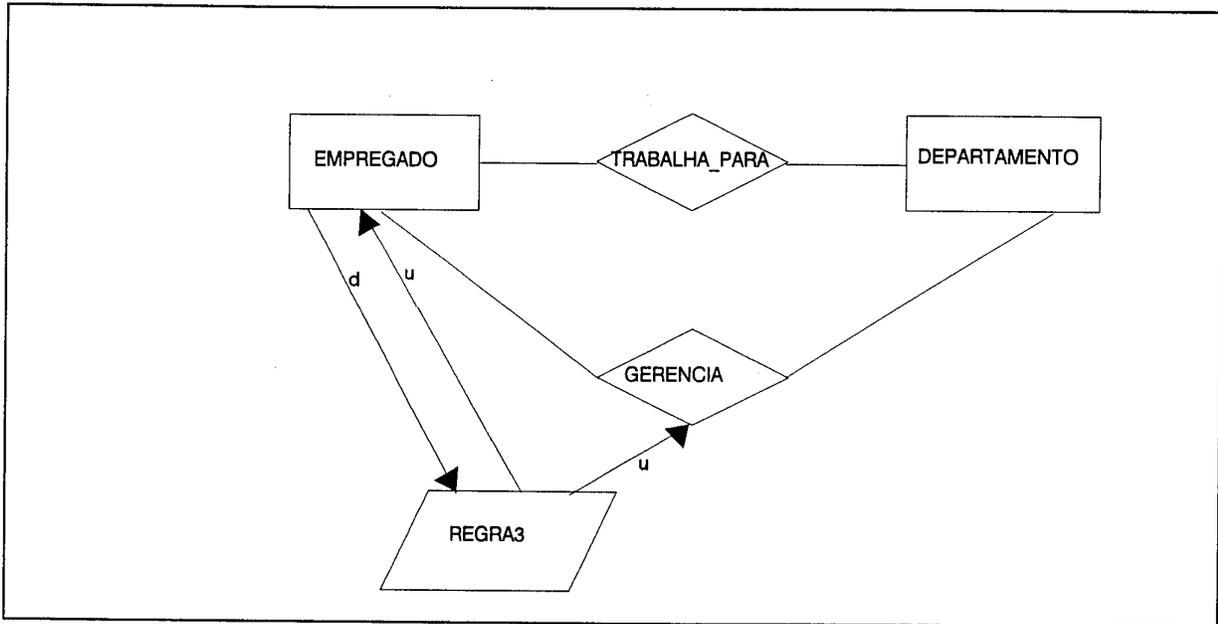


Figura 3.1  
Diagrama E-R acrescido de regras [TANA90]

feita graficamente, pelo desenho dos lugares condicionais com círculos inscritos em quadrados, denotando decisão, e transição de ações por barras grossas, denotando uma sequência de ações básicas. A figura 3.2 apresenta um modelo da rede Lugar-Transição representativa de uma regra.

De acordo com a proposta de Tanaka [TANA90,TANA91], cada regra tem um lugar representando o evento disparador ( $T$ ), o qual é o resultado do disparo de barras de detecção de eventos ( $D_1, \dots, D_n$ ), cujos lugares de entrada são eventos de entrada ( $I_{i1}, \dots, I_{mj}$ ).

De modo geral, um evento disparador será uma combinação de um número arbitrário de eventos de entrada, conectados por operadores do tipo AND e OR. Os eventos agrupados por operadores AND convergem para a mesma barra detetora de eventos e as barras detetoras de evento convergem para para o mesmo lugar denotando disjunção. O lugar do evento disparador  $T$  se conecta a cada uma das barras de ação, guardadas pelos correspondentes lugares de condição. Cada barra de ação se conecta ao conjunto associado de lugares de eventos resultantes, que são os eventos de saída da regra.

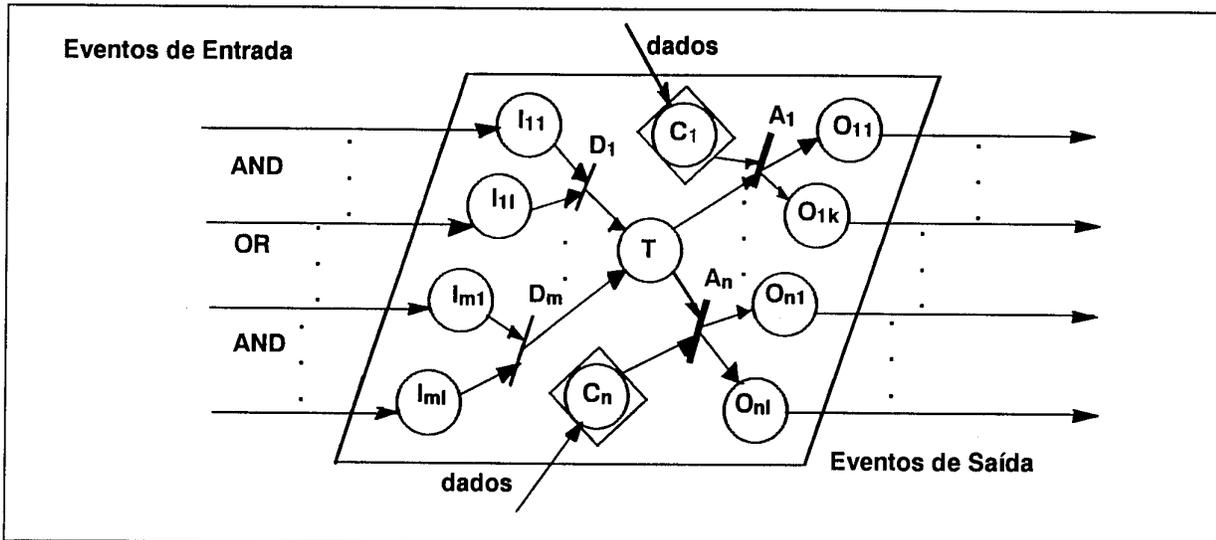


Figura 3.2  
Representação de uma regra em termos de uma Rede de Petri [TANA90]

Na proposta de Tanaka, são considerados dois tipos de marcas, representando instâncias e condições verificadas. Além da sinalização da ocorrência de eventos ou da avaliação de condições, marcas carregam informação na forma de atributos de eventos, usados na avaliação de condições e execução subsequente de ações. Essa seria a razão para que, na proposta, fossem as marcas chamadas de mensagens.

O disparo de uma barra de ação significa que as mensagens do evento disparador e os lugares de condição são transferidos para a barra e a ação que isso representa é executada e os eventos resultantes ocorrem. Esses eventos podem, por sua vez, tornar-se eventos de entrada de outras regras ou, mesmo, da própria regra considerada inicialmente.

A representação da natureza dinâmica do sistema inteiro é obtida pela junção dos lugares de condição e de eventos que são comuns nas redes representativas das regras singulares; ao grafo resultante é dado o nome de rede de regras e esta representa uma parte do processamento feito

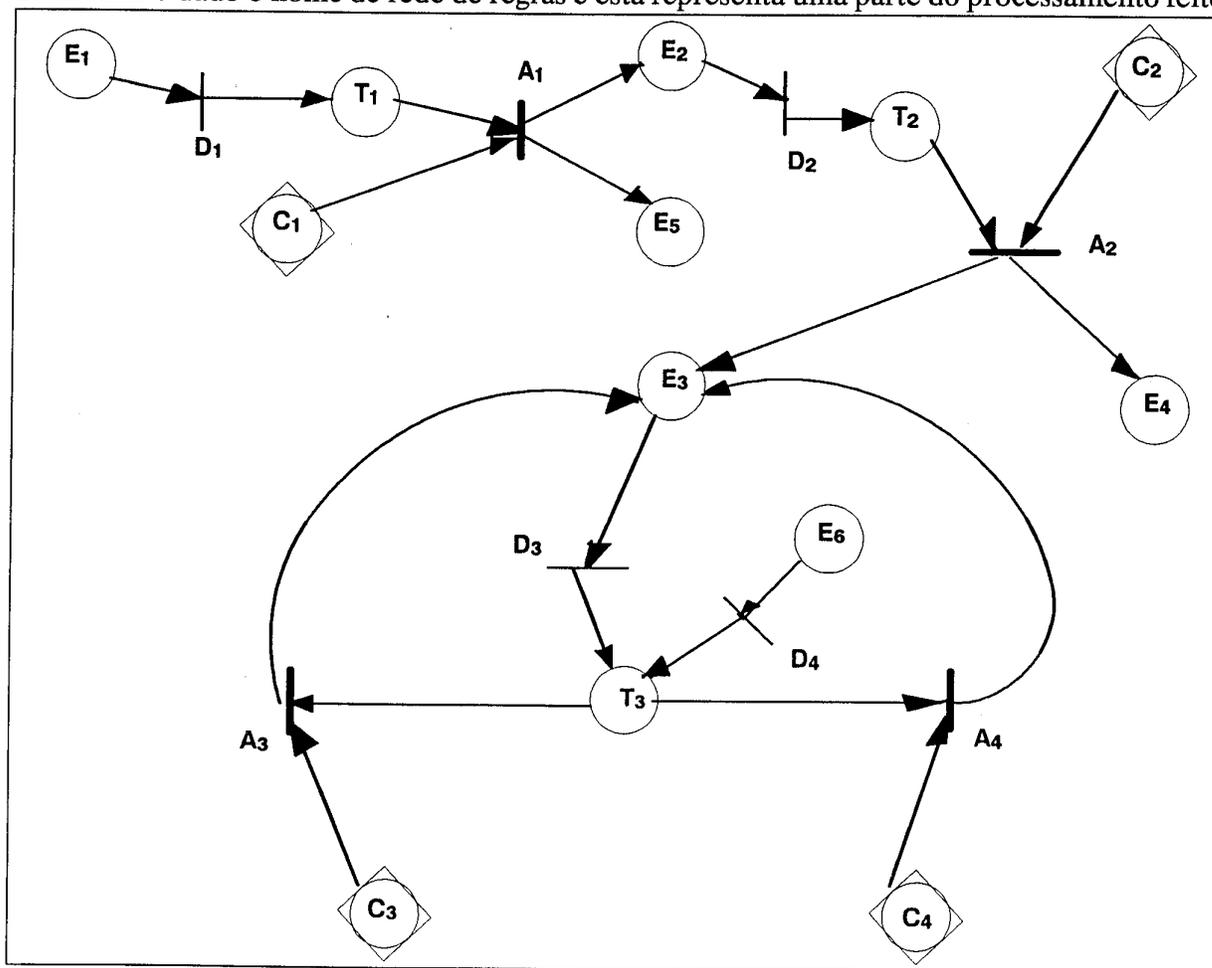


Figura 3.3  
 Rede de Petri correspondente às regras modeladas [TANA90]

no banco de dados, que é suportado por um SGBD ativo nos níveis lógico e físico. A figura 3.3 mostra uma Rede de Petri resultante da junção de várias redes de regras singulares.

## 4. PROPOSTA: O USO DE REDES COMPACTAS PARA MODELAGEM DE REGRAS

Nesta Seção apresentamos nossa proposta de representação de conceitos estáticos e dinâmicos com o uso de Redes Compactas Anotadas, introduzindo a Linguagem de Anotação empregada e fornecendo a interpretação associada aos símbolos com que construímos as mesmas.

### 4.1 - A Linguagem de Anotação

Uma Rede Compacta resulta de um processo de agregação de uma Rede Elementar envolvendo diversas instâncias de entidades de um mesmo tipo; a compreensão do seu funcionamento requer definição precisa das conexões e dos lugares elementares representados por cada elemento da Rede Compacta. Essa definição é feita em uma linguagem formal denominada Linguagem de Anotação (LA) [HEUS90], apresentada a seguir.

No caso de sistemas de informação, a LA é voltada para a descrição dos objetos abstraídos da realidade modelada, denominados elementos, e para a descrição de correspondências e propriedades envolvendo esses elementos. O conjunto de todos os elementos considerados no modelo compõe o UD desse modelo.

As descrições são compostas por expressões de dois tipos:

- fórmulas, que designam os valores lógicos FALSO e VERDADEIRO
- termos, que designam, cada um, uma entidade do UD.

A LA adotada pertence à classe definida por Heuser [HEUS90], baseada em Lógica de Primeira Ordem e acrescida de símbolos não lógicos para tratamento de conjuntos; sua sintaxe é apresentada na Tabela 4.1.

- *< símb. predicado >* ou *< símb. predicado binário >* é uma palavra, iniciada por letra maiúscula, contendo letras minúsculas e podendo usar *\_* (símbolo para sublinhado); pode, no caso do símbolo de predicado binário, ser um símbolo especial, como *<*, *>*, *=* e *<-->*.
- *< símb. função >* ou *< símb. função binária >* é uma palavra composta por letras maiúsculas e podendo usar *\_* (símbolo para sublinhado); pode, no caso do símbolo de função binária, ser um símbolo especial como *+*, *-*, *\** e *x*. Para evitar ambigüidades - caso de símbolos de função associados a tipos de atributo de nomes iguais mas relacionados a tipos de entidade ou de relacionamento diferentes - estendemos a sintaxe definida por Heuser. Assim, em casos do tipo o símbolo de função é formado por duas palavras separadas pelo símbolo *.* (ponto): a primeira delas é composta por letras maiúsculas e a segunda por letras minúsculas.
- *< var >* é uma palavra, composta apenas por letras minúsculas;
- *< constante >* é uma cadeia de caracteres entre apóstrofos ou um valor numérico; no caso de constantes que designam conjuntos de elementos do UD, seus nomes são iniciados por *C\_*.
- *POTEN* é o símbolo de função que associa a cada conjunto o seu conjunto potência. O conjunto potência é definido por:  
$$POTEN (conj) ::= \{ subconj \mid subconj \text{ Sub conj } \}$$
- *x* é o símbolo de função binária denotativo do Produto Cartesiano. Se *C\_Auto*, por exemplo, designa o conjunto de todos os automóveis e *C\_Cor* designa o conjunto de

todas as cores, então o termo ( $C\_Auto \times C\_Cor$ ) designa o conjunto de todos os pares que se pode formar com um automóvel e uma cor.

Para cada aplicação, a LA empregada deve se estendida com os símbolos de predicado e os símbolos de função específicos para a realidade modelada; a extensão é feita pela definição dos símbolos e dos números de argumentos respectivos (*aridade*).

<b>&lt; fórmula &gt; ::=</b>	<b>Verdadeiro   Falso   &lt; simb. predicado &gt; (&lt; termo &gt;, ...)</b> <b>  (&lt; termo &gt; &lt; simb. predicado binário &gt; &lt; termo &gt;)</b> <b>  (&lt; fórmula &gt; e &lt; fórmula &gt;)</b> <b>  (&lt; fórmula &gt; ou &lt; fórmula &gt;)  não &lt; fórmula &gt;</b> <b>  (&lt; fórmula &gt; impl &lt; fórmula &gt;)</b> <b>  paratodo &lt; var &gt; (&lt; fórmula &gt;)</b> <b>  existe &lt; var &gt; (&lt; fórmula &gt;)  (&lt; fórmula &gt;)</b>
<b>&lt; simb.predicado &gt; ::=</b>	<b>(de acordo com o modelo)</b>
<b>&lt; simb.predicado binário &gt; ::=</b>	<b>Elem   Sub   =  </b> <b>(outros de acordo com o modelo)</b>
<b>&lt; termo &gt; ::=</b>	<b>&lt; constante &gt;   &lt; var &gt;   &lt; simb.função &gt; (&lt; termo &gt;, ...)</b> <b>  (&lt; termo &gt; &lt; simb.função binária &gt; &lt; termo &gt;)</b> <b>  {&lt; termo &gt;, ...}   {&lt; var &gt;   &lt; fórmula &gt; }</b> <b>  &lt; &lt; termo &gt;, ... &gt;   (&lt; termo &gt;)</b>
<b>&lt; simb.função &gt; ::=</b>	<b>&lt; conj.potência &gt;   (outros de acordo com o modelo)</b>
<b>&lt; simb.função binária &gt; ::=</b>	<b>x   (outros de acordo com o modelo)</b>
<b>&lt; conj.potência &gt; ::=</b>	<b>POTEN &lt; termo &gt; =</b>

Tabela 4.1 - Sintaxe da LA [HEUS90]

## 4.2 - A Modelagem de Regras com Redes Compactas

Conforme abordado na Seção 2, a modelagem conceitual dos aspectos estáticos e dinâmicos de sistemas de informação pode ser feita com Redes de Petri; um tipo especial de Rede - a Rede Compacta - é empregado por Heuser [HEUS89,HEUS90], resultando em uma representação homogênea e com nível de formalismo adequado a sua utilização nas etapas subsequentes do processo de projeto de sistemas de informação.

A modelagem das regras que definem restrições de integridade dos dados modelados ou que descrevem reações do componente sistêmico responsável pela gerência dos dados em face de acontecimentos detectados no seu ambiente externo é considerada necessária desde as primeiras etapas do processo de projeto de sistemas de informação. Conforme abordado na Seção 3, tem sido buscada a especificação dessas regras desde a etapa de modelagem dos conceitos do domínio da aplicação, bem como a definição de correspondências nos níveis de abstração mais baixos (mapeamento de conceitos).

Neste trabalho, a proposta de modelagem de regras é uma alternativa à abordagem apresentada por Tanaka [TANA90,TANA91] no sentido de integração da representação de regras à modelagem dos demais aspectos dos sistemas de informação a nível conceitual e com o uso de Redes de Petri.

Assim, em lugar do emprego de Redes de Petri apenas a partir do nível lógico, propomos o uso de Redes de Petri Compactas na construção do modelo conceitual, produzindo um único modelo conceitual de dados, funções e regras observadas na realidade considerada.

#### 4.2.1 - A Detecção de Eventos

Em seu trabalho, Tanaka [TANA90, TANA91] utiliza lugares que recebem marcas denotativas da ocorrência de dois tipos de evento:

- eventos relevantes para o banco de dados representado, como alteração nos dados, o atingimento de uma data-limite ou uma ocorrência externa ao sistema;
- eventos disparadores de ações, definidos em termos da detecção de ocorrência dos eventos considerados relevantes para o banco de dados.

No nosso trabalho, as marcas denotativas da ocorrência de ambos os tipos de evento são guardadas em lugares de uma Rede Compacta e a cada lugar associamos uma anotação indicando, com termos da LA, o conjunto de elementos do UD que o mesmo pode conter.

#### 4.2.2 - A Representação de Ações

As ações são modeladas por alterações, agrupadas por Tanaka em dois tipos [TANA90,TANA91]:

- transições que representam a detecção de uma configuração particular de eventos e a conseqüente produção de um ou mais eventos disparadores;
- transições que representam ações executadas na presença de um evento disparador e na avaliação de uma sentença como VERDADEIRO.

Modelamos, em nosso trabalho, os tipos de transições através de conexões compactas representadas por retângulos anotados com a correspondente indicação da detecção ou ação modelada. Uma transição que, além da habilitação usual, também necessita que uma determinada sentença seja avaliada como VERDADEIRO tem inscrita ou anotada próxima do símbolo da conexão a fórmula correspondente, expressa na LA definida.

Associado a cada ramo de conexão e junto ao símbolo da conexão, anotamos na LA um termo designativo de um conjunto de entidades pertencentes ao domínio do lugar ao qual o ramo está ligado.

## 5. ESTUDO DE CASO

O exemplo considerado a seguir - esquema conceitual, regras e restrições de integridade - foi adaptado do trabalho de Tanaka [TANA91].

A figura 5.1 apresenta o modelo conceitual de um banco de dados, denominado de COMPANHIA, construído com o uso da abordagem E-R. De acordo com Tanaka, a visão apresentada é uma adaptação de uma parte do esquema para o banco de dados da COMPANHIA tal como apresentado por R. Elmasri e S. B. Navathe (referenciado por Tanaka como [ELMA89]).

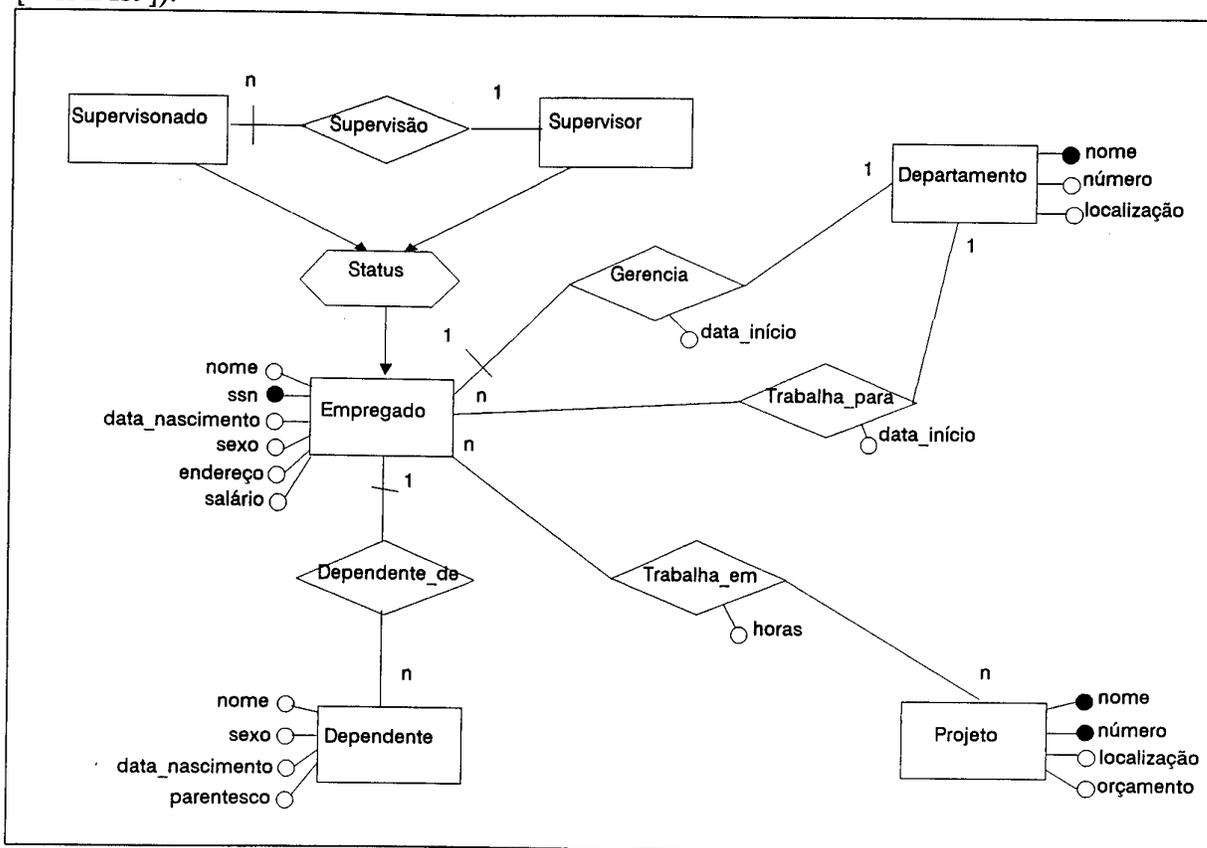


Fig. 5.1 - Modelo E-R para o BD Companhia (Adaptado de [TANA90])

Neste trabalho, o modelo apresentado por Tanaka foi adaptado através da alteração de alguns detalhes visando compatibilidade com as Regras 1 e 2, apresentadas na seção 5.1 juntamente com as demais, e facilitar a ilustração do uso de Redes de Petri:

a - modificamos a ação associada à Regra 1, fazendo com que EMPREGADO que trabalhe menos de 6 horas no PROJETO afetado seja desligado apenas desse PROJETO (em lugar de retirá-lo do banco de dados), dando origem a um novo evento\_resultante (EMPREGADO desligado);

b - criamos uma nova regra (Regra 4) com a política para demissão de um EMPREGADO desligado de um PROJETO; segundo essa regra, um EMPREGADO só é demitido quando não possui qualquer ligação com PROJETOS através do tipo de relacionamento TRABALHA\_EM;

c - incluímos o atributo DATA\_INÍCIO no relacionamento TRABALHA\_PARA. Acreditamos que esse atributo é necessário para viabilizar a política para substituição de um gerente demitido (Regra 2);

d - suprimimos o atributo NÚMERO DE EMPREGADOS da entidade DEPARTAMENTO, já que seus possíveis valores são derivados dos relacionamentos pertencentes ao tipo de relacionamento TRABALHA\_PARA, estabelecido entre os tipos de entidade DEPARTAMENTO e EMPREGADO;

e - consideramos atômico o tipo de atributo NOME do tipo de entidade EMPREGADO, em lugar de considerá-lo como uma tupla composta por PRIMEIRO\_NOME, NOME\_DO\_MEIO e NOME\_FINAL;

f - consideramos atômico o tipo de atributo LOCALIZAÇÕES do tipo de entidade DEPARTAMENTO, e passamos a designá-lo por LOCALIZAÇÃO;

g - na representação gráfica das entidades, atributos e relacionamentos utilizamos os símbolos e convenções adotados por Setzer [SETZ86] e empregados por Heuser [HEUS90], com o objetivo de apresentar os Diagramas E-R de uma forma única;

h - substituímos o auto-relacionamento SUPERVISÃO da entidade EMPREGADO por uma hierarquia semântica (generalização/especialização), com o objetivo de reduzir a ambigüidade da representação.

## 5.1 - Regras

Tanaka [TANA91] considera em seu trabalho as seguintes regras:

*Regra 1* (Política para redução de orçamento de um projeto)

*Evento disparador:* atualização de PROJETO.orçamento

*Condição:* o novo orçamento torna-se menor que 80% do velho orçamento

*Ação:* - desligar os EMPREGADOS que TRABALHAM\_EM o PROJETO afetado um número menor do que 6 horas/semana (*operação de eliminação* em TRABALHA\_EM);

*Eventos resultantes:* {EMPREGADO desligado, revisão salarial}.

Cabe observar que o conjunto dos empregados que trabalham menos de 6 horas em um projeto cujo orçamento tenha sido reduzido pode ser vazio, não produzindo o efeito de redução de custos pretendido.

Tanaka aponta para a possibilidade de "*revisão salarial*" ser um evento externo produzido pela regra, podendo ser um evento de entrada para outra regra ou iniciar um processo externo.

*Regra 2* (Política para substituição de um gerente demitido)

*Evento disparador:* EMPREGADO demitido

*Condição:* o empregado demitido estava gerenciando um departamento

*Ação:* - associar como novo gerente o EMPREGADO com o maior número de anos de serviço entre aqueles que TRABALHAM\_PARA o DEPARTAMENTO afetado;

- incrementar em 20% o salário do EMPREGADO designado novo gerente (*operação de alteração* em EMPREGADO)

*Eventos resultantes:* {GERENCIA modificado, EMPREGADO modificado}

*Regra 3 (Política de limitação de salário)*

*Evento\_disparador:* EMPREGADO alterado

*Condição:* EMPREGADO.salário menor que \$10.000

*Ação:* - atribuir aos salários inferiores a \$10.000 o valor \$10.000 (*operação\_de\_alteração* em EMPREGADO)

*Eventos\_resultantes:* {EMPREGADO alterado}

*Condição:* EMPREGADO.salário maior que \$100.000

*Ação:* - atribuir aos salários maiores que \$100.000 o valor \$100.000 (*operação\_de\_alteração* em EMPREGADO)

*Eventos\_resultantes:* {EMPREGADO alterado}

*Regra 4 (Política de demissão de EMPREGADO)*

*Evento\_disparador:* EMPREGADO demitido

*Condição:* EMPREGADO não TRABALHA\_EM pelo menos um PROJETO

*Ação:* - retirar o EMPREGADO (*operação\_de\_eliminação* em EMPREGADO)

*Eventos\_resultantes:* {EMPREGADO eliminado}

A integridade referencial pode ser mantida pela modelagem de regras aplicáveis às meta-entidades consideradas. Assim, por exemplo, se uma instância de EMPREGADO é eliminada do banco de dados, todas as instâncias de relacionamento em que participa, bem como todas as instâncias de entidades fracas com que se relaciona devem ser igualmente eliminadas do banco de dados (eliminação em cascata).

Conforme Tanaka [TANA90] observa, este tipo de regra para garantia da integridade referencial não é capturada no Diagrama E-R básico, sendo especificado como meta-conhecimento, isto é, como propriedade invariante da Abordagem E-R.

## **5.2 - A Representação do Modelo E-R como Rede de Petri**

Aplicamos sobre o modelo conceitual para o banco de dados COMPANHIA apresentado na figura 5.1 as regras de tradução do Modelo E-R para Redes de Petri, abordadas na Seção 2. Conforme podemos observar nas figuras 5.2 a 5.10, o modelo apresenta uma dificuldade natural para sua utilização, já que cada entidade, atributo ou relacionamento é traduzido por um lugar na Rede.

Para dominar a complexidade da apresentação desse esquema conceitual sob a forma de Rede de Petri recorreremos a duas abordagens simplificadoras, conforme exposto a seguir.

Inicialmente, identificamos os conjuntos de elementos do UD que sofrem modificação e aqueles que permanecem inalterados, considerando para isso a dinâmica do sistema modelado [HEUS90]. Em seguida, restringimos essa dinâmica àquela decorrente apenas da execução das regras descritas na seção 5.1. Finalmente, identificamos os conjuntos que permanecem fixos e aqueles cujo conteúdo varia durante o funcionamento do sistema de regras com o objetivo de definir aqueles que serão representados como lugares na Rede e os que serão traduzidos por expressões da LA. A especificação desses conjuntos é abordada na seção 5.3.

Essa simplificação utilizada no exemplo não deve ser considerada aplicável em grau semelhante a outras situações. No caso de projetos para situações reais, de complexidade mais elevada, a simplificação seria feita a partir das restrições de integridade definidas pela realidade considerada como objeto da aplicação.

Como segunda forma simplificadora, transformamos o Modelo E-R em várias Redes de

Petri, cada uma correspondendo a uma visão parcial dos conjuntos de entidades e de relacionamentos do banco de dados considerado. As Redes assim obtidas são apresentadas nas figuras 5.11 a 5.13. Embora essa forma de segmentação não seja apresentada por Heuser [HEUS90] explicitamente como uma técnica para domínio da complexidade de problemas do tipo, acreditamos que a mesma obedece aos mesmos princípios por ele empregados no estudo de alternativas para utilização de Redes de Petri na modelagem conceitual, isoladamente ou em conjunto com Diagramas E-R.

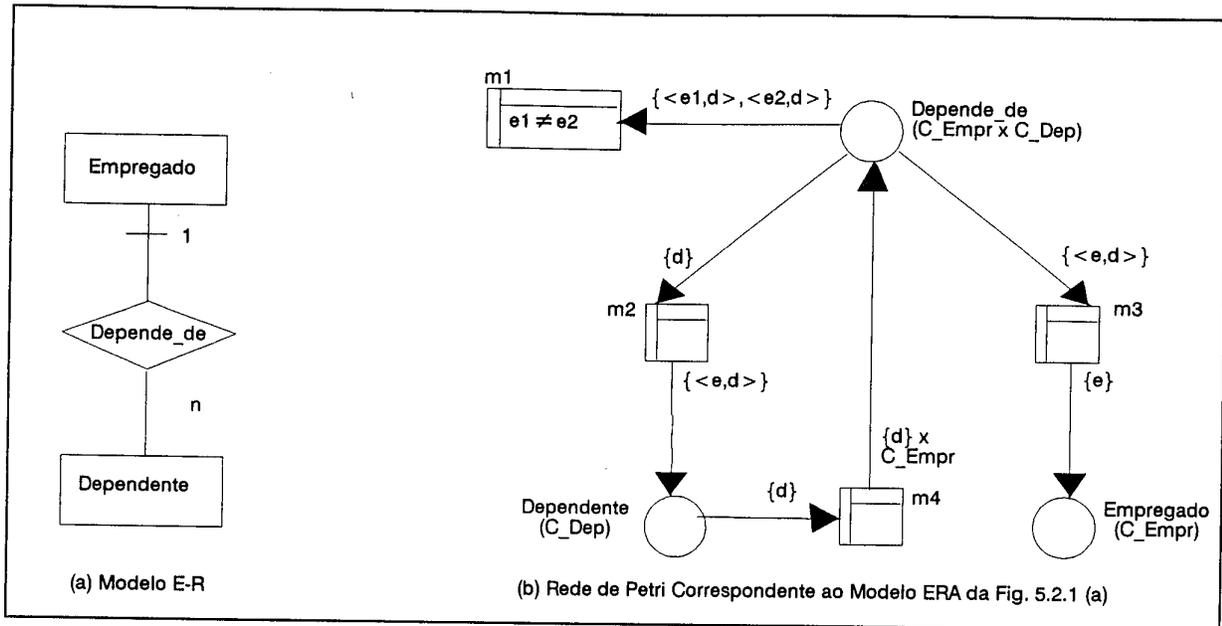


Figura 5.2  
Tradução para Rede de Petri do Relacionamento entre as Entidades Empregado e Dependente

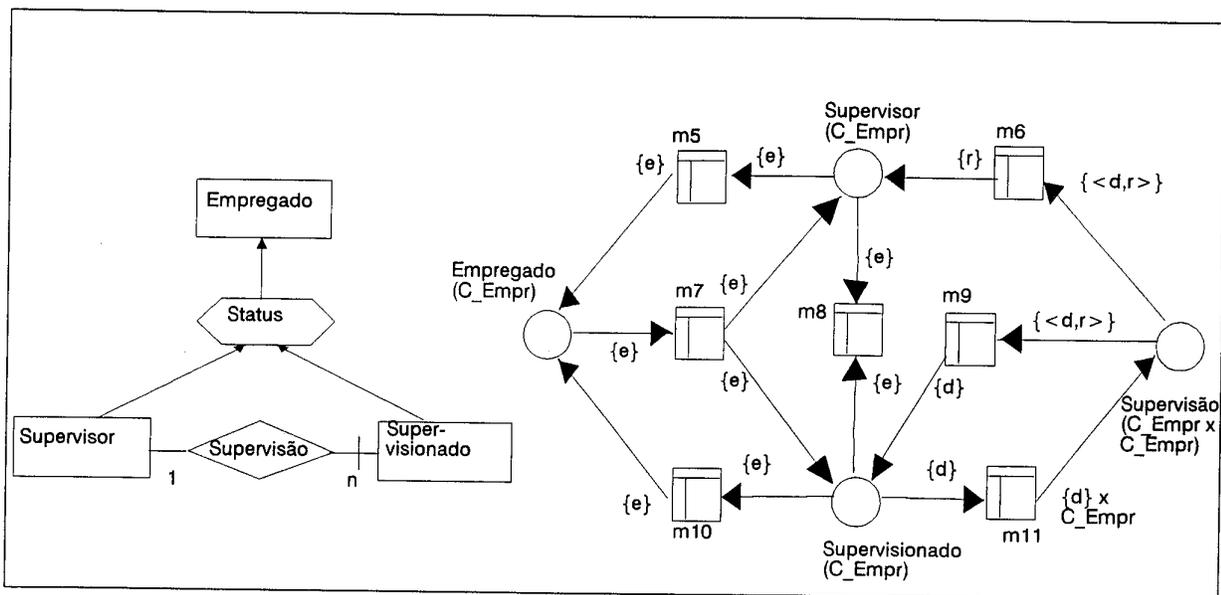


Figura 5.3  
Tradução para Rede de Petri da Especialização/Generalização de Empregado e do Relacionamento Supervisão

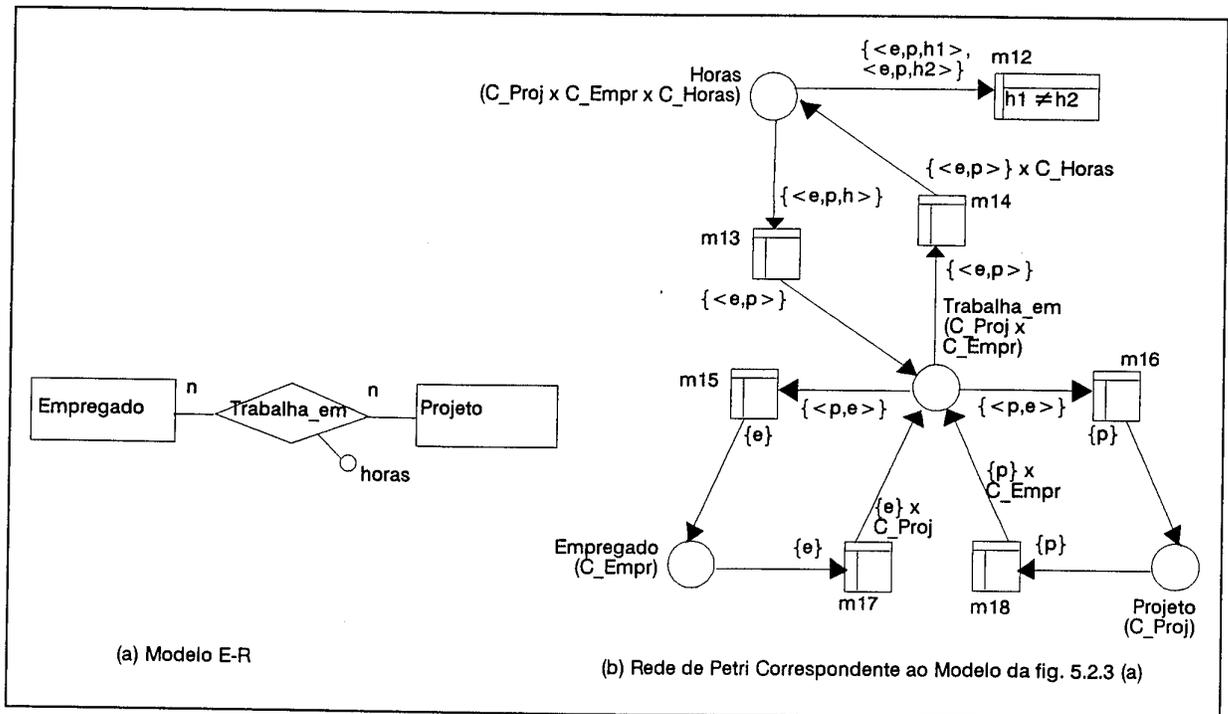


Figura 5.4 Tradução para Rede de Petri do Relacionamento entre as Entidades Empregado e Projeto

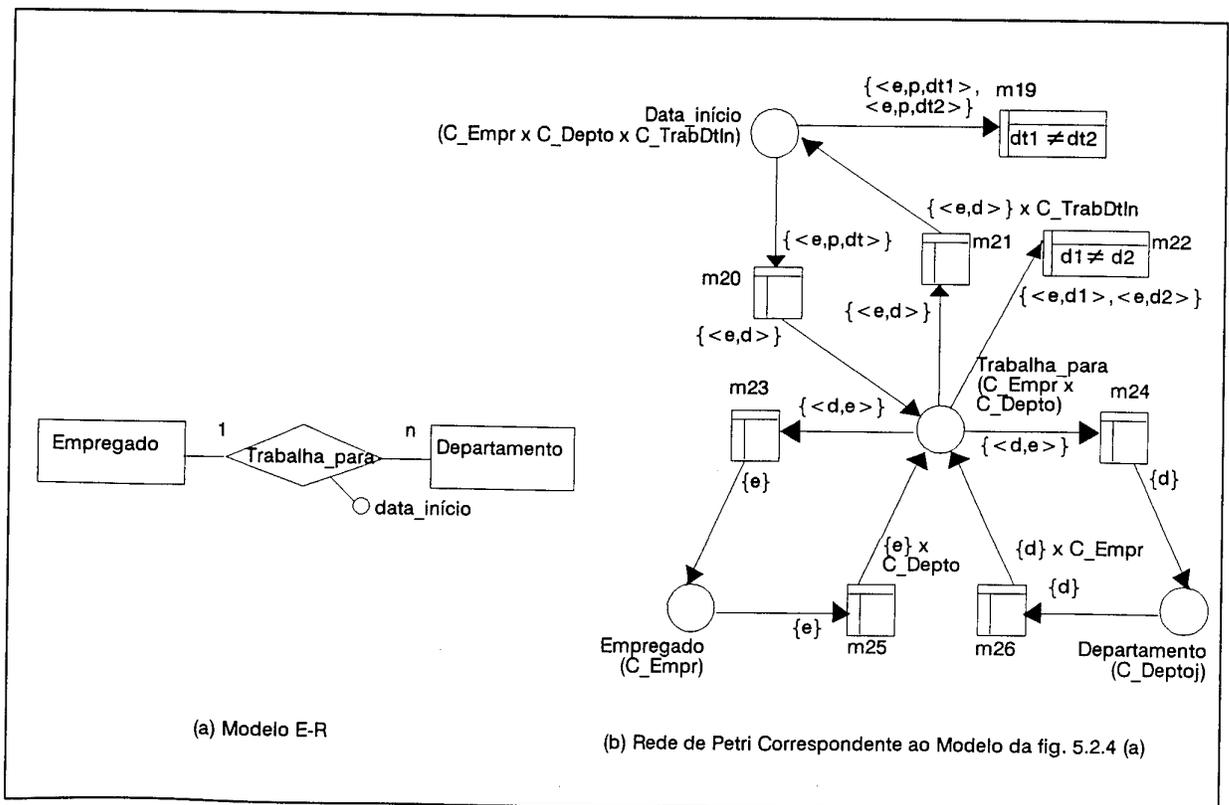
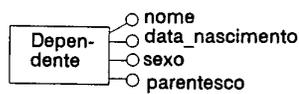
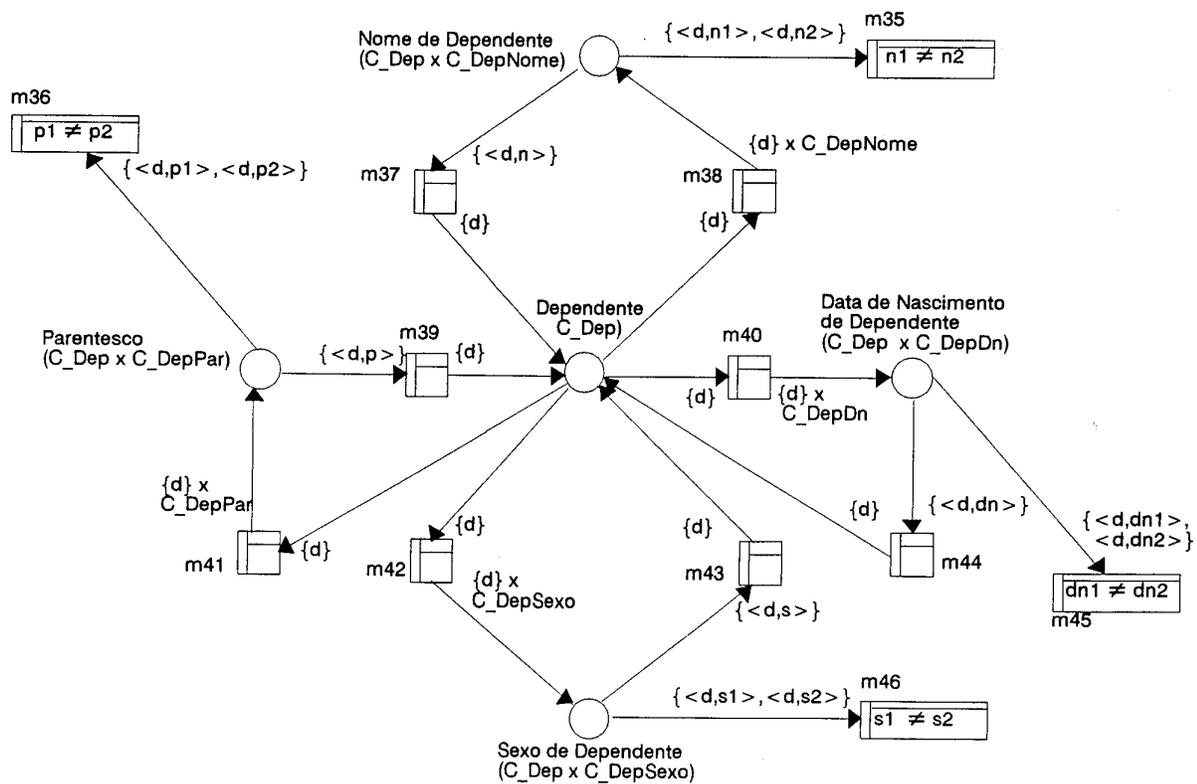


Figura 5.5 Tradução para Rede de Petri do Relacionamento Trabalha\_para entre as Entidades Empregado e Departamento



(a) Modelo E-R



(b) Rede de Petri Correspondente ao Modelo E-R da fig. 5.2.6 (a)

Figura 5.6  
Tradução para Rede de Petri da Entidade Dependente e seus Atributos

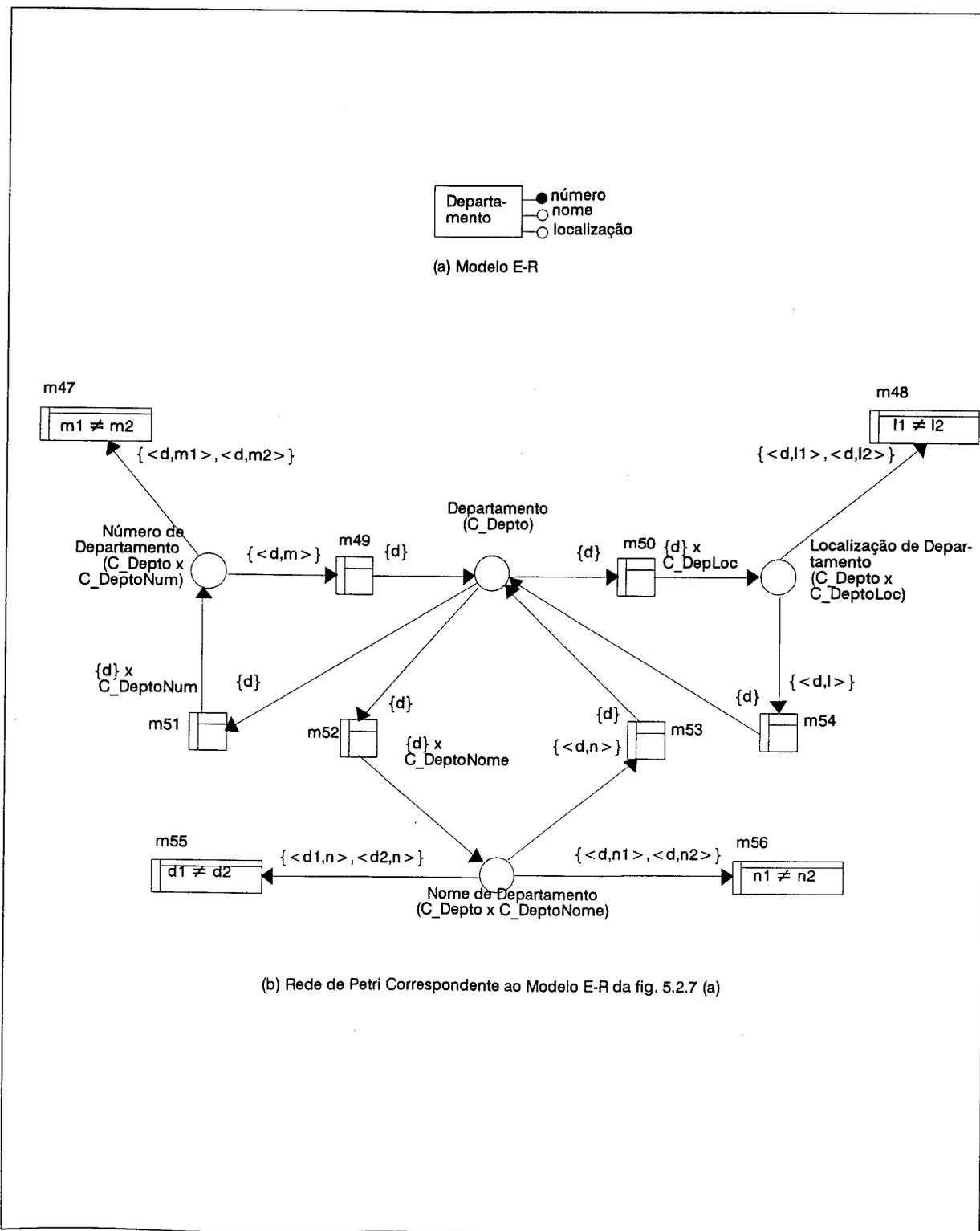
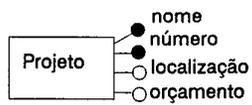
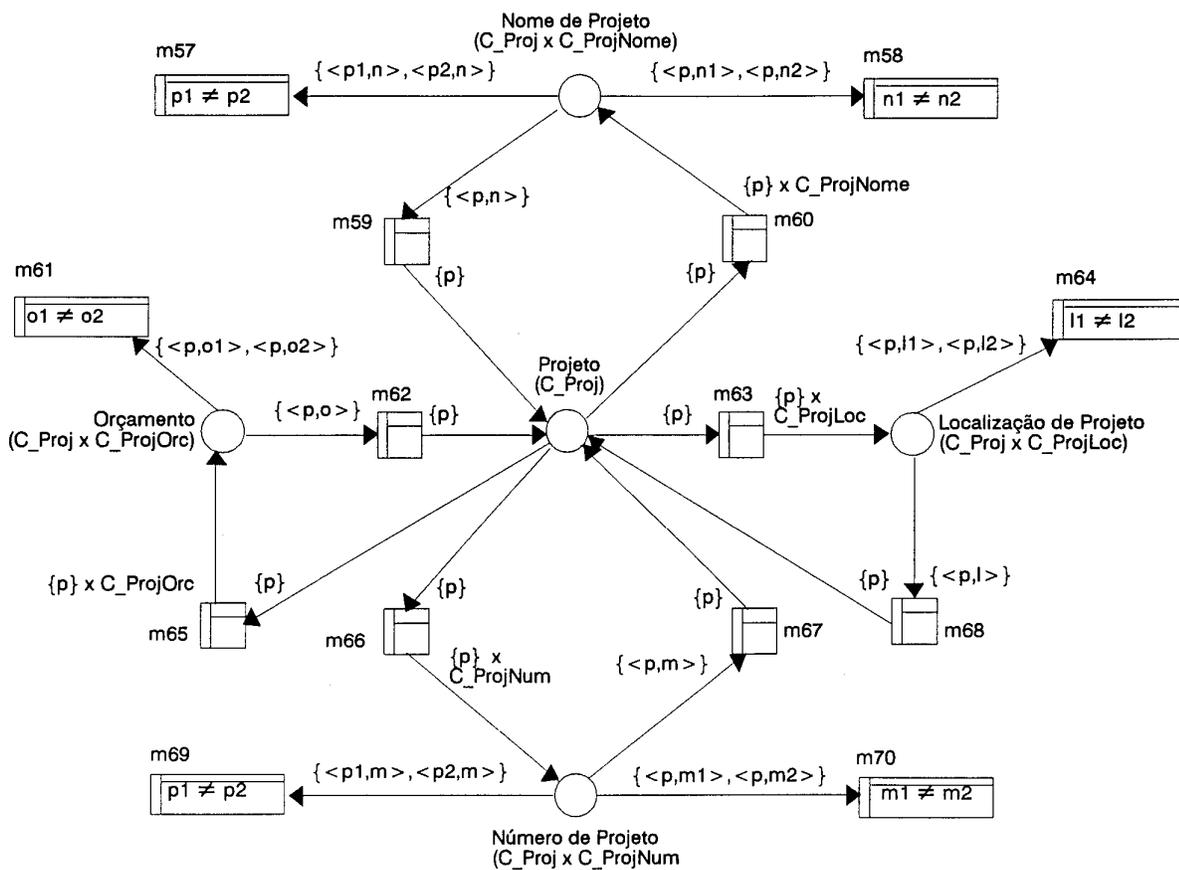


Figura 5.7  
Tradução para Rede de Petri da Entidade Departamento e seus Atributos



(a) Modelo E-R



(b) Rede de Petri Correspondente ao Modelo E-R da fig. 5.2.8 (a)

Figura 5.8  
Tradução para Rede de Petri da Entidade Projeto e seus Atributos

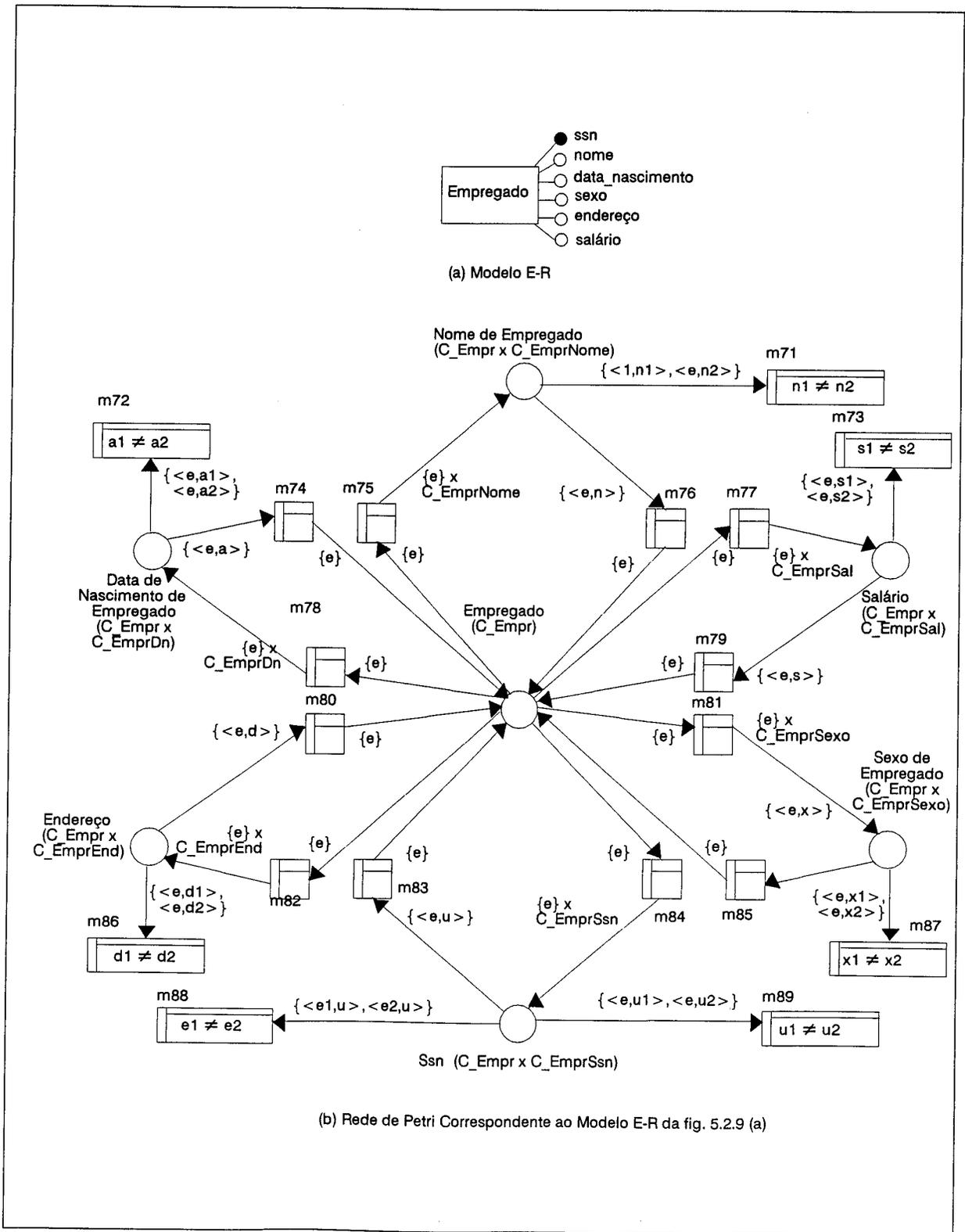


Figura 5.9  
Tradução para Rede de Petri da Entidade Empregado e seus Atributos

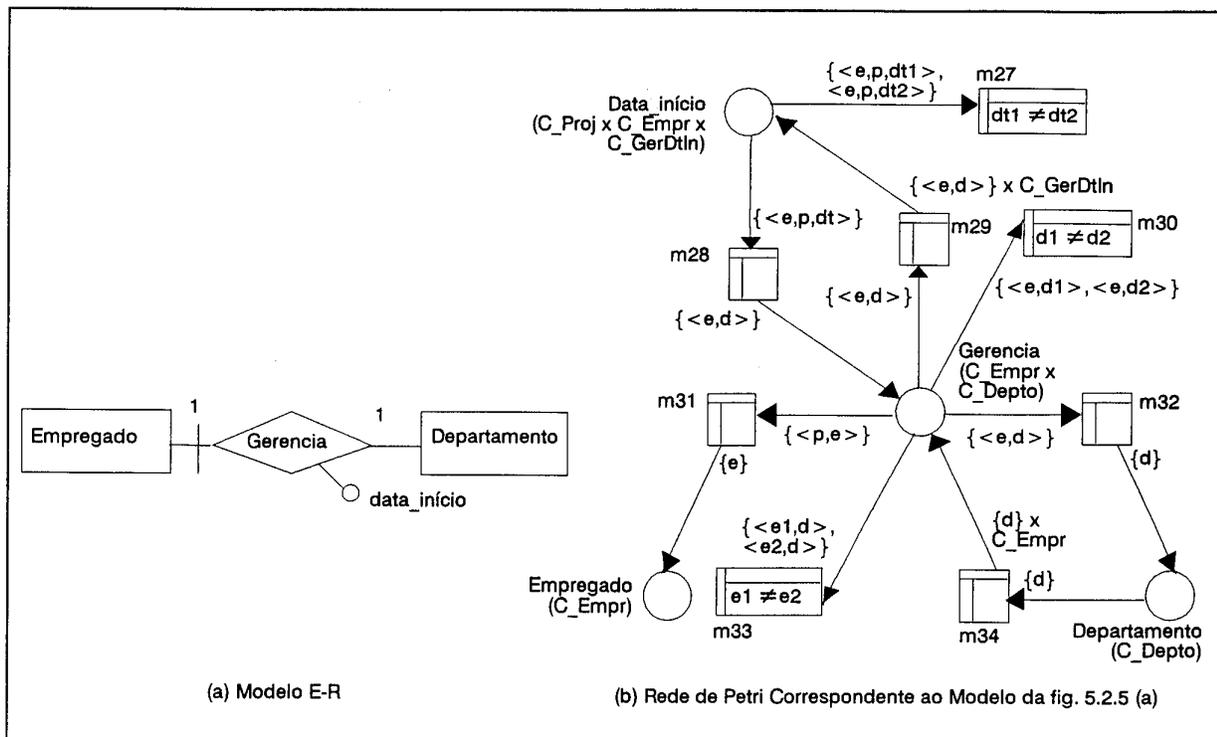


Figura 5.10

Tradução para Rede de Petri do Relacionamento Gerencia entre as Entidades Empregado e Departamento

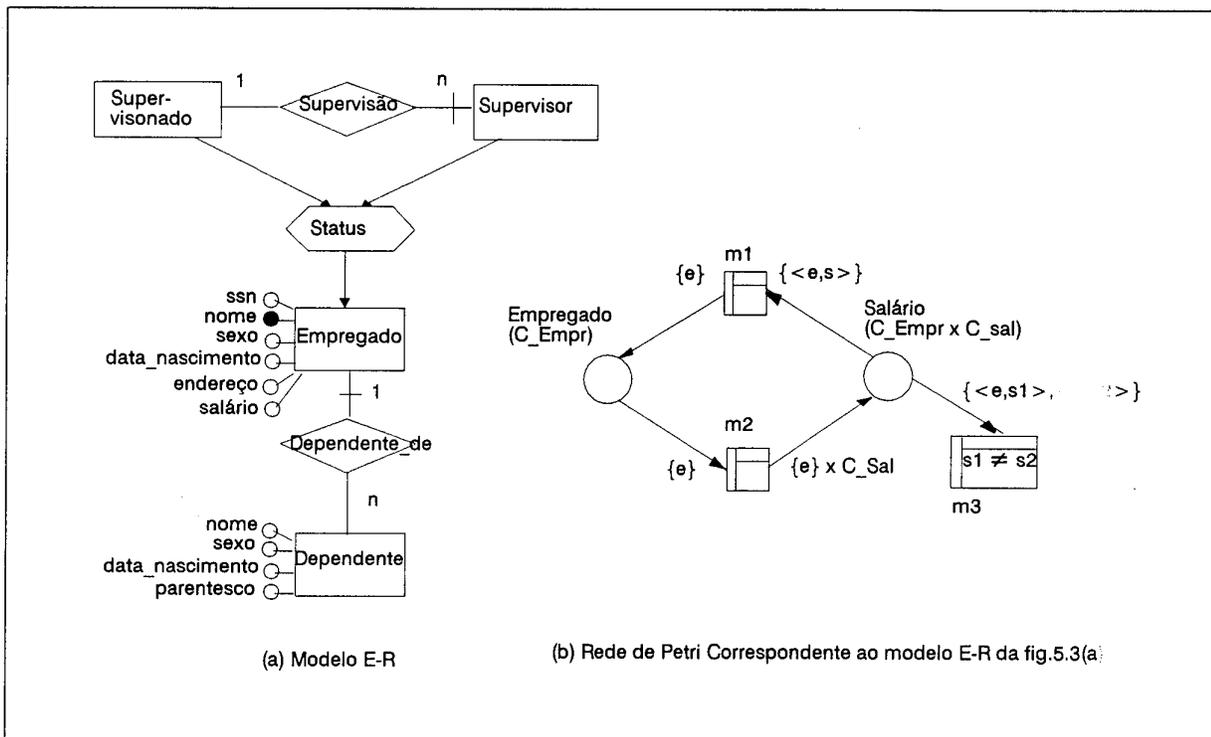
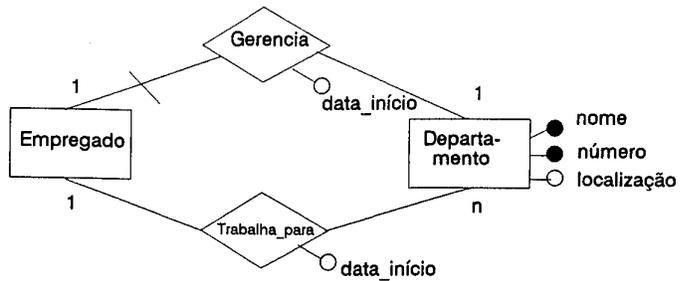
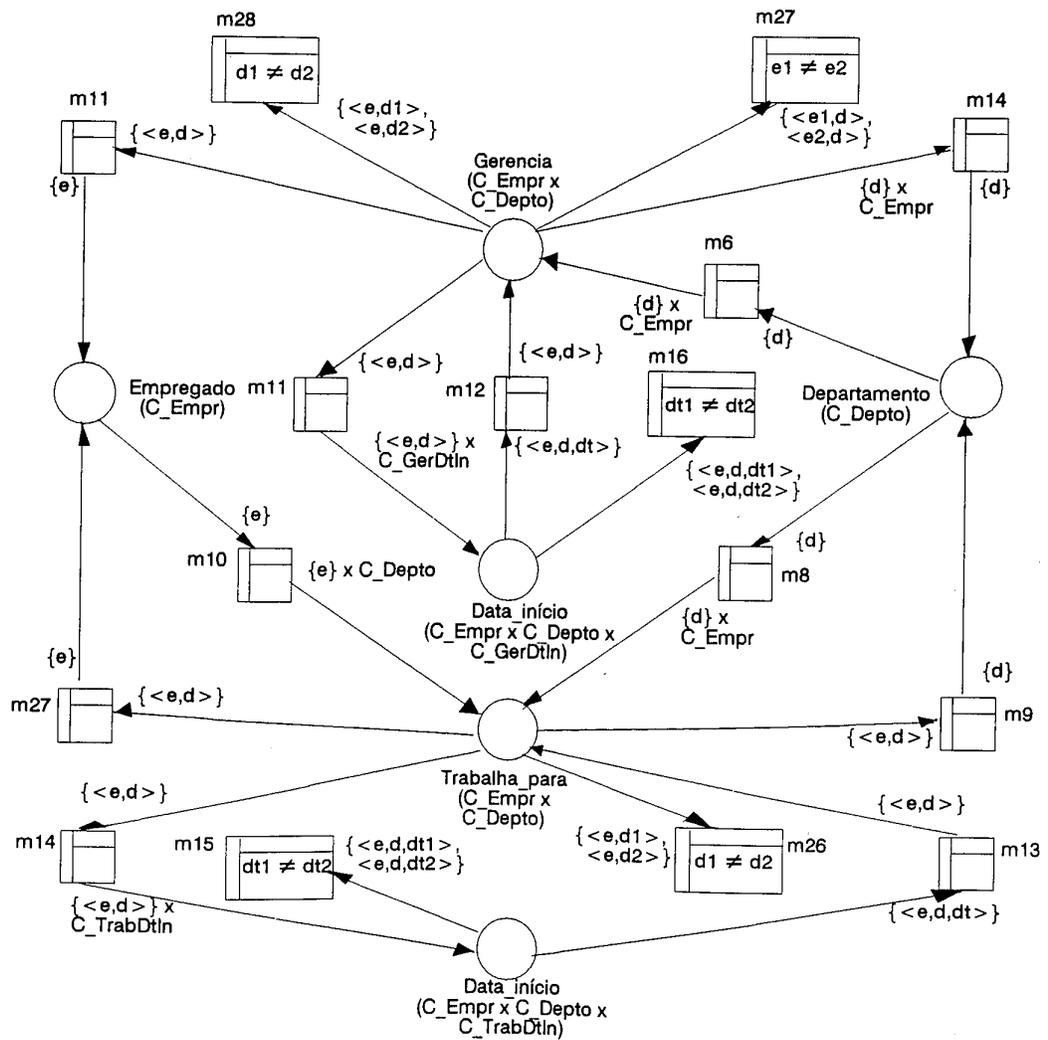


Figura 5.11 - Tradução para Rede de Petri das Entidades Empregado, Dependente e de seus Relacionamentos

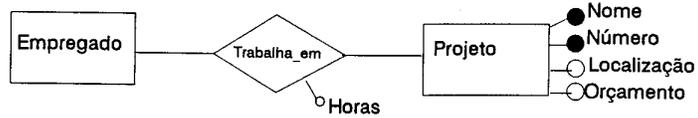


(a) Modelo E-R

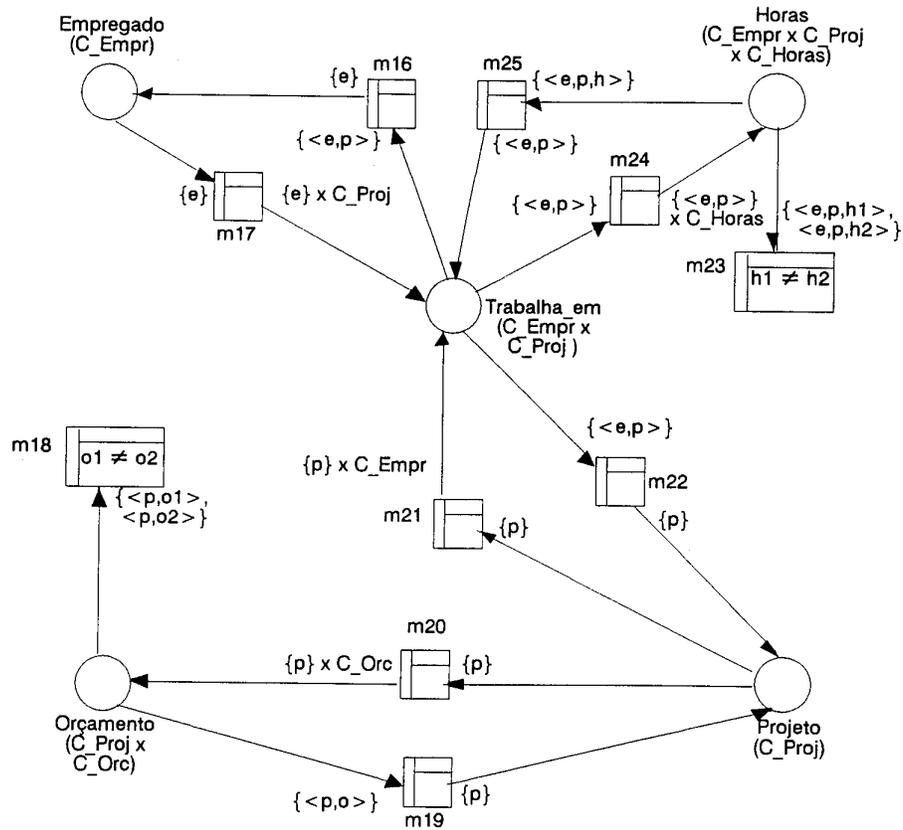


(b) Rede de Petri Correspondente ao Modelo E-R da fig. 5.4 (a)

Figura 5.12 - Tradução para Rede de Petri da Entidade Departamento e de seus Relacionamentos



(a) Modelo E-R



(b) Rede de Petri Correspondente ao Modelo E-R da fig. 5.5 (a)

Figura 5.13 - Tradução para Rede de Petri da Entidade Projeto e seu Relacionamento.

### 5.3 - Definição Parcial do UD (Conjuntos Constantes)

Heuser [HEUS90] representa cada conjunto de entidades de um Diagrama E-R por um lugar cujo domínio é definido por  $C\_Ent$ , conjunto de todas as entidades do UD, já que, na sua visão, um Diagrama E-R não define especificamente o domínio de um lugar.

Neste trabalho, para caracterização do domínio de um lugar representativo de um tipo de entidade modelada no Diagrama E-R, definimos o conjunto correspondente e sua relação com o conjunto de todas as entidades do Diagrama E-R e o conjunto de todos os atributos do Diagrama E-R.

O conjunto de todos os tipos de entidade do Diagrama E-R pode ser definido, utilizando os nomes que os identificam como

$$C\_Tipo\_Entidade = \{ "EMPREGADO", "DEPENDENTE", "PROJETO", "DEPARTAMENTO" \}.$$

O conjunto de todos os tipos de relacionamento entre as entidades do Diagrama E-R considerado pode ser definido, utilizando os nomes que os identificam, como

$$C\_Tipo\_Relacionamento = \{ "GERENCIA", "TRABALHA\_PARA", "TRABALHA\_EM", "SUPERVISÃO", "DEPENDENTE\_DE" \}.$$

Segundo Chen [CHEN76], um atributo pode ser definido como uma função que mapeia, em cada instante, um tipo de entidade ou um tipo de relacionamento em um conjunto de valores ou em um produto cartesiano de valores. O conjunto de valores caracteriza o tipo de atributo.

Trata-se, portanto, de um mapeamento dinâmico o qual, no processo de tradução do Modelo E-R para um Modelo CEM, deve ficar representado pela rede e não pela LA.

Entretanto, associado a cada tipo de entidade (relacionamento) há, no Modelo E-R, um conjunto constante, isto é, independente do tempo, constituído pelos nomes que identificam seus atributos. A utilização desses conjuntos constantes de nomes de atributos permite-nos, na linguagem de anotação, distinguir, usando termos distintos, os domínios dos diversos tipos de entidades e relacionamentos. Isso evita o uso indistinto do termo  $C\_Ent$  (que designa o conjunto de todos os elementos do UD) para designar os domínios dos lugares que representam, na Rede CEM, os diferentes tipos de entidade presentes no Modelo E-R.

Neste trabalho, com a finalidade específica de distinguir esses diferentes domínios, usaremos o nome do tipo de atributo (eventualmente qualificado pelo nome do tipo de entidade ou tipo de relacionamento ao qual esteja associado). O conjunto de todos os nomes de tipos de atributo do Diagrama E-R pode ser definido como:

$$C\_Nome\_de\_Atributo = \{ "DEPARTAMENTO.nome", "DEPARTAMENTO.número", "DEPARTAMENTO.localização", "DEPENDENTE.nome", "DEPENDENTE.sexo", "DEPENDENTE.data\_de\_nascimento", "parentesco", "EMPREGADO.nome", "ssn", "EMPREGADO.data\_de\_nascimento", "EMPREGADO.sexo", "endereço", "salário", "PROJETO.nome", "PROJETO.número", "PROJETO.localização", "orçamento", "horas", "TRABALHA\_PARA.data\_inicio", "GERENCIA.data\_inicio" \}$$

Podemos, então, definir as funções:

$$ATRIBENT : C\_tipo\_ENTIDADE \rightarrow POTEN(C\_Nome\_de\_Atributo)$$
$$ATRIBREL : C\_tipo\_RELACIONAMENTO \rightarrow POTEN(C\_Nome\_de\_Atributo)$$

e obter

$$ATRIBENT("DEPARTAMENTO") = C\_Atrib\_DEPARTAMENTO = \{ "DEPARTAMENTO.nome", "DEPARTAMENTO.número", "DEPARTAMENTO.localização" \}.$$

**ATRIBENT("DEPENDENTE") = C\_Atrib\_DEPENDENTE =**  
 = {"DEPENDENTE.nome", "DEPENDENTE.sexo", "DEPENDENTE.data\_de\_nascimento", "parentesco"}.

**ATRIBENT("EMPREGADO") = C\_Atrib\_EMPREGADO =**  
 = {"EMPREGADO.nome", "EMPREGADO.sexo", "EMPREGADO.data\_de\_nascimento", "ssn", "endereço", "salário"}.

**ATRIBENT("PROJETO") = C\_Atrib\_PROJETO =**  
 = {"PROJETO.nome", "PROJETO.número", "PROJETO.localização", "orçamento"}.

**ATRIBREL("TRABALHA\_EM") = C\_Atrib\_TRABALHA\_EM = {"horas"}**

**ATRIBREL("TRABALHA\_PARA") = C\_Atrib\_TRABALHA\_PARA = {"TRABALHA\_PARA.data\_início"}**

**ATRIBREL("GERENCIA") = C\_Atrib\_GERENCIA = {"GERENCIA.data\_início"}**

**ATRIBREL("SUPERVISÃO") = C\_Atrib\_SUPERVISÃO = { }**

**ATRIBREL("DEPENDENTE\_DE") = C\_Atrib\_DEPENDENTE\_DE = { }.**

Finalmente, devemos ainda definir os conjuntos:

- C\_Entidade, constituído por todas as constantes da LA que designam *entidades* (instâncias dos tipos de entidade) do UD;
- C\_Relacionamento, constituído por todas as constantes da LA que designam *relacionamentos* (instâncias dos tipos de relacionamento) do UD.

Podemos, agora, definir os domínios dos lugares que modelam os tipos de entidade presentes no Diagrama E-R considerado no exemplo:

C\_Dep é o conjunto de todos os imagináveis dependentes, definido pelo tipo de entidade DEPENDENTE; formalmente:

**C\_Dep = {dep | dep Elem C\_Entidade e ATRIBENT(dep) = C\_Atrib\_DEPENDENTE}.**

C\_Depto é o conjunto de todos os imagináveis departamentos, definido pelo tipo de entidade DEPARTAMENTO:

**C\_Depto = {depto | depto Elem C\_Entidade e ATRIBENT(depto) = C\_Atrib\_DEPARTAMENTO}.**

C\_Empr é o conjunto de todos os imagináveis empregados, definido pelo tipo de entidade EMPREGADO, formalmente:

**C\_Empr = {empr | empr Elem C\_Entidade e ATRIBENT(empr) = C\_Atrib\_EMPREGADO}.**

C\_Proj é o conjunto de todos os imagináveis projetos, definido pelo tipo de entidade PROJETO; formalmente:

**C\_Proj = {proj | proj Elem C\_Entidade e ATRIBENT(proj) = C\_Atrib\_PROJETO}.**

C\_EmprNome é o conjunto de todos os possíveis nomes atribuíveis a uma pessoa empregada (conceito intuitivo).

C\_EmprSsn é o conjunto de todos os possíveis números atribuíveis a um empregado (conceito intuitivo).

C\_EmprDn é o conjunto de todas as possíveis datas\_de\_nascimento das pessoas empregadas (conceito intuitivo).

C\_EmprSexo é o conjunto dos possíveis valores de sexo de uma pessoa empregada; formalmente:

$C\_EmprSexo = \{\text{"masculino"}, \text{"feminino"}\}.$

C\_EmprEnd é o conjunto de todos os possíveis endereços atribuíveis a uma pessoa empregada (conceito intuitivo).

C\_EmprSal é o conjunto de todos os possíveis valores de salário atribuíveis a um empregado (conceito intuitivo).

C\_DepNome é o conjunto de todos os possíveis nomes atribuíveis a uma pessoa dependente de empregado (conceito intuitivo).

C\_DepSexo é o conjunto dos possíveis valores de sexo de uma pessoa dependente de empregado ; formalmente:

$C\_DepSexo = \{\text{"masculino"}, \text{"feminino"}\}.$

C\_DepDn é o conjunto de todas as possíveis datas\_de\_nascimento das pessoas dependentes de empregado (conceito intuitivo).

C\_DepPar é o conjunto de todos os possíveis parentescos entre as pessoas empregadas e seus dependentes (conceito intuitivo).

C\_DeptoNome é o conjunto de todos os possíveis nomes atribuíveis a um departamento (conceito intuitivo).

C\_DeptoNum é o conjunto de todos os possíveis números atribuíveis a um departamento (conceito intuitivo).

C\_DeptoLoc é o conjunto de todas as possíveis localizações atribuíveis a um departamento (conceito intuitivo).

C\_ProjNome é o é o conjunto de todos os possíveis nomes atribuíveis a um projeto (conceito intuitivo).

C\_ProjNum é o conjunto de todos os possíveis números atribuíveis a um projeto (conceito intuitivo).

C\_ProjLoc é o conjunto de todas as possíveis localizações atribuíveis a um projeto (conceito intuitivo)

C\_ProjOrc é o conjunto de todos os possíveis valores de orçamento atribuíveis a um projeto (conceito intuitivo)

C\_Horas é o conjunto de todos os possíveis números de horas por semana trabalhadas por uma pessoa em um projeto (conceito intuitivo).

C\_GerDtIn é o conjunto de todas as possíveis datas em que uma pessoa empregada pode iniciar a gerência de um departamento (conceito intuitivo).

C\_TrabDtIn é o conjunto de todas as possíveis datas em que uma pessoa empregada pode começar a trabalhar para um departamento (conceito intuitivo).

#### **5.4 - Definição da LA**

A LA utilizada na anotação das Redes de Petri obtidas a partir da transformação do Modelo E-R do banco de dados COMPANHIA foi definida, a partir da classe proposta em [HEUS90], acrescentando símbolos de função e símbolos de relação, conforme apresentado a seguir. Cabe observar que foram considerados para a definição dos símbolos de função todos os atributos não afetados pelos eventos alteradores especificados, não sendo, portanto, representados por lugares na Rede.

#### 5.4.1- Símbolos de Função

EMPREGADO.nome é o símbolo da função que associa a uma entidade do tipo EMPREGADO um valor pertencente ao domínio do tipo de atributo NOME; formalmente:

**EMPREGADO.nome: C\_Empr --> C\_EmprNome.**

EMPREGADO.data\_de\_nascimento é o símbolo da função que associa a uma entidade do tipo EMPREGADO um valor pertencente ao domínio do tipo de atributo DATA\_DE\_NASCIMENTO; formalmente:

**EMPREGADO.data\_de\_nascimento: C\_Empr --> C\_EmprDn.**

EMPREGADO.sexo é o símbolo da função que associa a uma entidade do tipo EMPREGADO um valor pertencente ao domínio do tipo de atributo SEXO; formalmente:

**EMPREGADO.sexo: C\_Empr --> C\_EmprSexo.**

ENDEREÇO é o símbolo da função que associa a uma entidade do tipo EMPREGADO um valor pertencente ao domínio do tipo de atributo ENDEREÇO; formalmente:

**ENDEREÇO: C\_Empr --> C\_EmprEnd.**

DEPENDENTE.nome é o símbolo da função que associa a uma entidade do tipo DEPENDENTE um valor pertencente ao domínio do tipo de atributo NOME; formalmente:

**DEPENDENTE.nome: C\_Dep --> C\_DepNome.**

DEPENDENTE.data\_de\_nascimento é o símbolo da função que associa a uma entidade do tipo DEPENDENTE um valor pertencente ao domínio do tipo de atributo DATA\_DE\_NASCIMENTO; formalmente:

**DEPENDENTE.data\_de\_nascimento: C\_Dep --> C\_DepDn.**

DEPENDENTE.sexo é o símbolo da função que associa a uma entidade do tipo DEPENDENTE um valor pertencente ao domínio do tipo de atributo SEXO; formalmente:

**DEPENDENTE.sexo: C\_Dep --> C\_DepSexo.**

PARENTESCO é o símbolo da função que associa a uma entidade do tipo DEPENDENTE um valor pertencente ao domínio do tipo de atributo PARENTESCO; formalmente:

**PARENTESCO: C\_Dep --> C\_DepPar.**

DEPARTAMENTO.nome é o símbolo da função que associa a uma entidade do tipo DEPARTAMENTO um valor pertencente ao domínio do tipo de atributo NOME; formalmente:

**DEPARTAMENTO.nome: C\_Depto --> C\_DeptoNome.**

DEPARTAMENTO.número é o símbolo da função que associa a uma entidade do tipo DEPARTAMENTO um valor pertencente ao domínio do tipo de atributo NÚMERO; formalmente:

**DEPARTAMENTO.número: C\_Depto --> C\_DeptoNum.**

DEPARTAMENTO.localização é o símbolo da função que associa a uma entidade do tipo DEPARTAMENTO um valor pertencente ao domínio do tipo de atributo LOCALIZAÇÃO; formalmente:

**DEPARTAMENTO.localização: C\_Depto --> C\_DeptoLoc.**

PROJETO.nome é o símbolo da função que associa a uma entidade do tipo PROJETO um valor pertencente ao domínio do tipo de atributo NOME; formalmente:

**PROJETO.nome: C\_Proj --> C\_ProjNome.**

PROJETO.número é o símbolo da função que associa a uma entidade do tipo PROJETO um valor pertencente ao domínio do tipo de atributo NÚMERO; formalmente:

**PROJETO.número: C\_Proj --> C\_ProjNum.**

PROJETO.localização é o símbolo da função que associa a uma entidade do tipo PROJETO um valor pertencente ao domínio do tipo de atributo LOCALIZAÇÃO; formalmente:

**PROJETO.localização: C\_Proj --> C\_ProjLoc.**

GERENCIA.data\_início é o símbolo da função que associa a uma tupla de entidades  $\langle e, d \rangle$  ( $e \in \text{Elem } C\_Empr$  e  $d \in \text{Elem } C\_Dept$ ) pertencente ao domínio do tipo de relacionamento GERENCIA um valor pertencente ao domínio do tipo de atributo DATA\_INÍCIO; formalmente:

**GERENCIA.data\_início : (C\_Empr x C\_Depto) --> C\_GerDtIn.**

TRABALHA\_PARA.data\_início é o símbolo da função que associa a uma tupla de entidades  $\langle e, d \rangle$  ( $e \in \text{Elem } C\_Empr$  e  $d \in \text{Elem } C\_Dept$ ) pertencente ao domínio do tipo de relacionamento TRABALHA\_PARA um valor pertencente ao domínio do tipo de atributo DATA\_INÍCIO; formalmente:

**TRABALHA\_PARA.data\_início : (C\_Empr x C\_Depto) --> C\_TrabDtIn.**

DEPTO\_GERIDO é o símbolo da função que associa a uma entidade do tipo de entidade EMPREGADO a entidade do tipo de entidade DEPARTAMENTO com a qual se relaciona através do relacionamento GERENCIA:

**DEPTO\_GERIDO: C\_Empr ---> C\_Dept.**

#### 5.4.2 - Símbolos de Predicado

**Mais\_antigo**, símbolo de predicado unário definido sobre o conjunto  $(C\_Empr \times C\_Depto)$ , definida por:

*paratodo e paratodo d (Mais\_antigo ( $\langle e, d \rangle$ ) <--> ( e Elem C\_Empr e d Elem C\_Depto e Trabalha\_para ( $\langle e, d \rangle$ ) e paratodo x (x Elem C\_Empr e Trabalha\_para ( $\langle x, d \rangle$ ) e não (x = e) --> (TRABALHA.data\_início( $\langle e, d \rangle$ ) Precede TRABALHA.data\_início( $\langle x, d \rangle$ ))))).*

onde o predicado unário **Precede** definido sobre o conjunto  $C\_TrabDtIn$  tem interpretação intuitiva.

**Trabalha\_para**, símbolo do predicado unário definido sobre o conjunto  $(C\_Empr \times C\_Depto)$ , definida por

*paratodo e paratodo d (Trabalha\_para( $\langle e, d \rangle$ ) <--> ( e Elem C\_Empr e d Elem C\_Depto e  $\langle e, d \rangle \in \text{Elem } (C\_Empr \times C\_Depto)$ )).*

**Trabalha\_em**, símbolo do predicado unário definido sobre o conjunto  $(C\_Empr \times C\_Proj)$ , definida por

*paratodo e paratodo p (Trabalha\_em( $\langle e, p \rangle$ ) <--> ( e Elem C\_Empr e p Elem C\_Proj e  $\langle e, p \rangle \in \text{Elem } (C\_Empr \times C\_Proj)$ )).*

#### 5.5 - A Rede Compacta

A figura 5.6 apresenta a Rede de Petri correspondente às regras definidas para o banco de dados COMPANHIA.

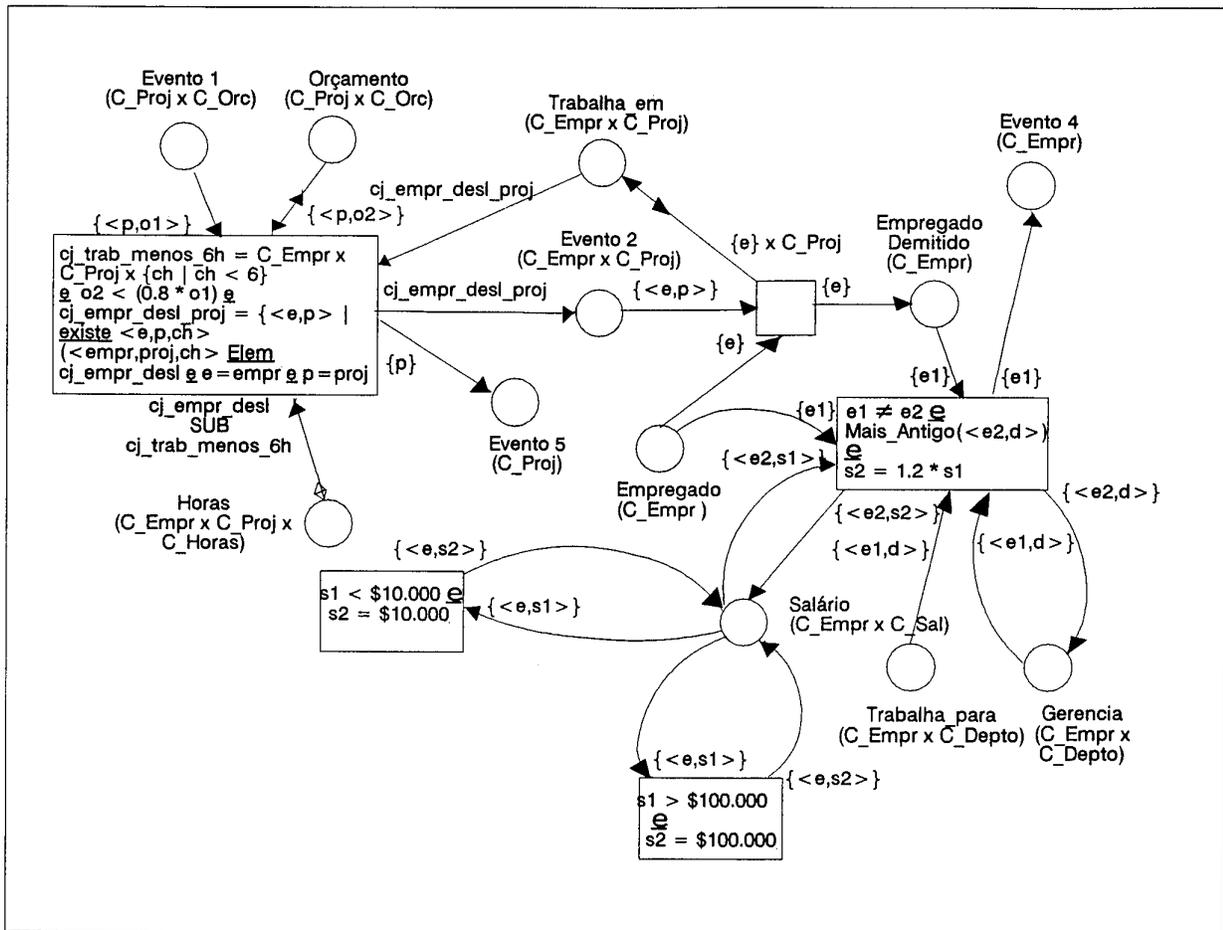


Figura 5.14 - Rede de Petri Para as Regras do BD Companhia

## 6. CONCLUSÃO

Apresentamos uma aplicação da proposta de Heuser [HEUS90] de representação conceitual dos aspectos estáticos e dinâmicos de bancos de dados para sistemas de informação com o uso de Redes de Petri Compactas. O estudo de caso apresentado permite uma análise comparativa com a abordagem apresentada por Tanaka [TANA90,TANA91].

Tanaka propõe o uso de Redes de Petri Elementares para a produção do modelo de implementação das regras que dizem respeito às entidades e relacionamentos representados no Modelo Conceitual, segundo o Modelo E-R. Neste trabalho, adotamos uma abordagem alternativa, empregando Redes de Petri Compactas para a produção de um Modelo Conceitual que contém a representação simultânea das propriedades estáticas e dinâmicas e que possui características que o tornam utilizável também como modelo lógico para implementação.

O uso de Redes de Petri Compactas possibilita representações de menores dimensões que as obtidas com Redes de Petri Elementares. Ainda assim, a representação conceitual das propriedades estáticas com o uso de conexões mortas pode gerar Redes de grandes dimensões, exigindo que se apliquem critérios de segmentação para o domínio da sua complexidade.

Como em outras abordagens que utilizam representações gráficas, o trabalho envolvido é grande e faz-se sentir a necessidade de suporte por ferramentas automatizadas específicas que facilitem o processo de construção do modelo.

## 6. CONCLUSÃO

Apresentamos uma aplicação da proposta de Heuser [HEUS90] de representação conceitual dos aspectos estáticos e dinâmicos de bancos de dados para sistemas de informação com o uso de Redes de Petri Compactas. O estudo de caso apresentado permite uma análise comparativa com a abordagem apresentada por Tanaka [TANA90,TANA91].

Tanaka propõe o uso de Redes de Petri Elementares para a produção do modelo de implementação das regras que dizem respeito às entidades e relacionamentos representados no Modelo Conceitual, segundo o Modelo E-R. Neste trabalho, adotamos uma abordagem alternativa, empregando Redes de Petri Compactas para a produção de um Modelo Conceitual que contém a representação simultânea das propriedades estáticas e dinâmicas e que possui características que o tornam utilizável também como modelo lógico para implementação.

O uso de Redes de Petri Compactas possibilita representações de menores dimensões que as obtidas com Redes de Petri Elementares. Ainda assim, a representação conceitual das propriedades estáticas com o uso de conexões mortas pode gerar Redes de grandes dimensões, exigindo que se apliquem critérios de segmentação para o domínio da sua complexidade.

Como em outras abordagens que utilizam representações gráficas, o trabalho envolvido é grande e faz-se sentir a necessidade de suporte por ferramentas automatizadas específicas que facilitem o processo de construção do modelo.

## BIBLIOGRAFIA

- [BERT91] Bertino E., e Martino, L., Object-Oriented Database Management System, in IEEE Computer, April 1991.
- [CHEN76] Chen, P.P., The Entity-Relationship Model - Towards a Unified View of Data, ACM, Transactions on Database Systems, Vol. 1, No.1, 1976.
- [HEUS89] Heuser, C.A., e Peres, E.M., Rumo a Um Modelo Conceitual Completo: Redes de Petri e Diagramas E/R, in Revista de Informática Teórica e Aplicada, Outubro de 1989, Vol. 1 Num. 1.
- [HEUS90] Heuser, C.A., Modelagem Conceitual de Sistemas, V Escola Brasileiro-Argentina de Informática, 1990.
- [MELO88] Melo, R.N., Bancos de Dados não Convencionais: a Tecnologia de BD e suas Novas Áreas de Aplicação, VI Escola de Computação, Campinas-SP, 1988.
- [OBER86] Oberquelle, H., Human-Machine Interaction and Role/Function/Action-Nets, in Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets, Berlin, 1986.
- [REIS86] Reising, W., Petri Nets in Software Engineering, in Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets, Berlin, 1986.
- [RICH85] Richter, G., e Voss, K., Towards a Comprehensive Model Integrating Information and Resources, in Advances in Petri Nets, Berlin, 1985.
- [SETZ86] Setzer, V.W., Projeto Lógico e Projeto Físico de Bancos de Dados, V Escola de Computação, Belo Horizonte, 1986.
- [TANA90] Tanaka, A.K., et al., Toward Conceptual Modeling of Active Databases, Technical Report TR-90-21, University of Florida, July 27, 1990.
- [TANA91] Tanaka, A.K., et al., Toward Conceptual Modeling of Active Databases, Update to the Technical Report TR-90-21, University of Florida, May 1991.
- [VOSS86a] Voss, K., Nets in Data Bases, in Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets, Berlin, 1986.
- [VOSS86b] Voss, K., Nets in Office Automation, in Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets, Berlin, 1986.
- [ZDON90] Zdonik, B., e Maier, D., Fundamentals of Object-Oriented Databases, in Readings in Object-Oriented Database Systems, Morgan Kaufmann Pub., Inc., California, 1990.