

**Aplicações Práticas de Inteligência Artificial
para Modelagem Conceitual de Banco de Dados**

Geoflávia Guilarducci de Alvarenga

Technical Report - RT-INF_001-05 - Relatório Técnico
October - 2005 - Outubro

The contents of this document are the sole responsibility of the authors.
O conteúdo do presente documento é de única responsabilidade dos autores.

**Instituto de Informática
Universidade Federal de Goiás**
www.inf.ufg.br

Aplicações Práticas de Inteligência Artificial para Modelagem Conceitual de Banco de Dados

Geoflavia Guilarducci de Alvarenga

geoflavia@inf.ufg.br

Abstract. The conceptual modeling phase is essential in database project process, because it represents the real needs business information system. A technique quite used in that phase is the ER model. Their concepts and characteristics are discussed, as well as, practical applications of Artificial Intelligence especially Knowledge Representation area, are related to the ER model's extension.

Keywords: Database Conceptual Modeling, Database Design, Entity-Relationship Model, Knowledge Representation in Database Conceptual Modeling.

Resumo. A fase de modelagem conceitual é essencial dentro de um processo de projeto de banco de dados, pois representa as necessidades reais do negócio de um sistema de informação. Uma técnica bastante usada nessa fase é o modelo ER. Seus conceitos e características são discutidos, assim como, aplicações práticas de Inteligência Artificial especialmente a área da Representação do Conhecimento, que estão relacionados à extensão do modelo ER.

Palavras-Chave: Modelagem Conceitual de Banco de Dados, Projeto de Banco de Dados, Modelo de Entidade Relacionamento, Representação do Conhecimento em Modelagem Conceitual de Banco de Dados.

1 Introdução

A demanda pelo processamento de informações tem aumentado bastante ao longo dos anos, e para atender essas demandas crescentes, é necessário soluções otimizadas e eficientes. Um sistema gerenciador de banco de dados (SGBD) se propõe a ajudar nessa tarefa, tornando-a mais ágil e fácil para os usuários interessados. Porém, para que o SGBD possa realizar bem essa tarefa, ele requer um planejamento prévio para realizar suas atividades, isto é, um processo de projeto de banco de dados precisa ser executado, objetivando modelar as estruturas que compartilharão as informações que serão processadas, armazenadas e mantidas por uma aplicação de um determinado sistema de informação de uma organização.

O processo de projeto de banco de dados pode conter algumas fases. Uma dessas fases mais importantes é a modelagem conceitual de banco de dados, pois é nessa fase que se procura elaborar uma visão integrada de todos os dados e processos de uma organização, totalmente independente do sistema computacional que será utilizado para solucionar os problemas dessa organização.

Um modelo de dados conceitual de alto nível bastante popular, o modelo ER, pode ser usado na fase de modelagem conceitual para modelar as estruturas e restrições de um banco de dados. Porém, na sua versão original, proposta em [1], viu-se a necessidade de agregar mais conceitos para modelar de maneira mais exata algumas aplicações complexas. Diante dessa necessidade, conceitos e técnicas de representação do conhecimento da Inteligência Artificial foram estudados, e contribuíram de maneira prática na adição de novos conceitos abstratos ao modelo ER.

No restante deste documento estão descritas as características fundamentais da modelagem conceitual de banco de dados (Seção 2), as aplicações práticas de Inteligência Artificial para modelagem conceitual de banco de dados (Seção 3) e as considerações finais do trabalho (Seção 4).

2 Modelagem Conceitual de Banco de Dados

A Figura 1 mostra uma descrição simplificada das principais fases do projeto de banco de dados. [5]

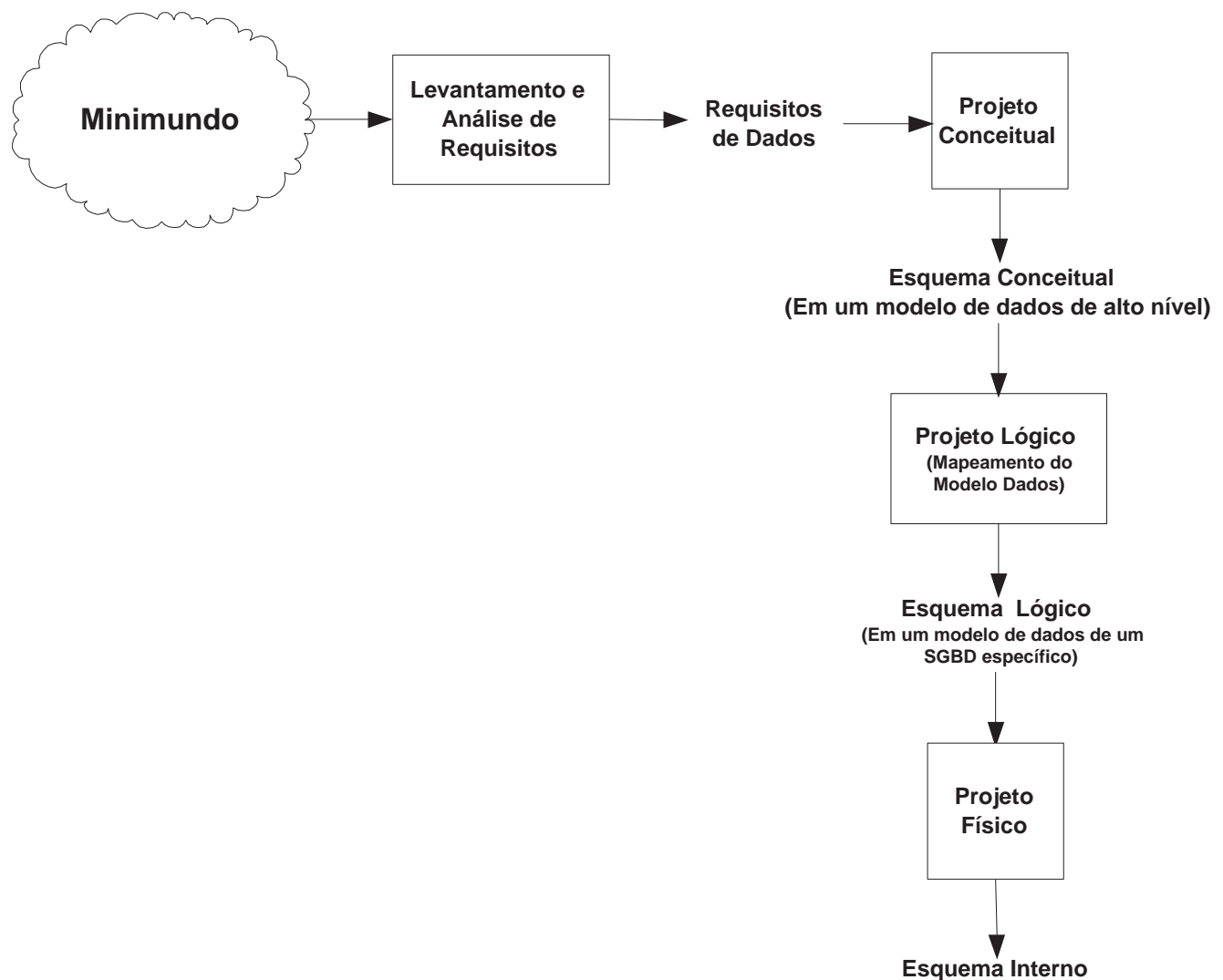


Figura 1: Diagrama simplificado das fases do projeto de um banco de dados.

Os passos do processo mostrado na Figura 1 [5], são detalhados a seguir:

- O primeiro passo é o levantamento e análise de requisitos. Durante essa fase, o projetista entrevista os possíveis usuários do banco de dados para entender e documentar seus requisitos de dados. O resultado dessa fase é o registro preciso e mais completo possível dos requisitos do usuário.
- Depois que os requisitos foram levantados e analisados, o próximo passo é criar um esquema conceitual para o banco de dados usando um modelo de dados conceitual (ou de alto nível). Essa fase é chamada projeto conceitual. O esquema conceitual é uma descrição dos requisitos de dados dos usuários, expressos através de conceitos fornecidos pelo modelo de dados de alto nível. Esses conceitos não incluem detalhes de implementação em computador, portanto, normalmente, eles são mais fáceis de serem compreendidos e podem ser utilizados na comunicação com os usuários não-técnicos. O esquema conceitual de alto nível também pode ser usado como uma referência para assegurar que todos os requisitos de dados do usuário sejam atendidos. Essa abordagem permite que os projetistas de banco de dados se concentrem na especificação das propriedades do dado, sem se preocupar com os detalhes de armazenamento.
- A próxima fase é a implementação real do banco de dados usando um Sistema Gerenciador de Bancos de Dados (SGBD) comercial. A grande maioria dos SGBDs comerciais atuais usa modelo de dados de implementação (modelo relacional ou modelo de banco de dados objeto-relacional) de maneira que o esquema conceitual seja transformado de um modelo de dados de alto nível em um modelo de dados de implementação. Essa fase é definida como projeto lógico.
- O último passo é a fase do projeto físico, durante a qual são definidas as estruturas de armazenamento interno, índices, caminhos de acesso e organização de arquivo para os arquivos do banco de dados

Conforme explicações do processo de projeto de banco de dados, podemos afirmar que a modelagem conceitual é uma etapa muito importante no planejamento de uma aplicação de um banco de dados bem-sucedido, pois através de um modelo de dados conceitual é possível identificar conceitos que descrevem os dados (da organização e dos negócios) sob a percepção e interpretação dos usuários, além disso, são mais simples de serem compreendidos por usuários leigos em informática [5, 7].

A participação e a compreensão do usuário nessa etapa é fundamental, pois é ele quem melhor conhece a organização e, portanto, aquele que melhor conhece os requisitos que a aplicação deve atender. Seu envolvimento é uma boa prática, porque aumenta a qualidade do software produzido [7].

A técnica de modelagem conceitual de banco de dados mais divulgada e usada é o modelo Entidade-Relacionamento (ER). Essa técnica foi criada por Peter Chen em 1976 [1, 2]. Ela pode ser considerada como um padrão de fato para modelagem conceitual [7].

O modelo ER pode ser usado como uma base para uma visão unificada dos dados. Esse modelo visualiza o mundo real através de **entidades** e seus **relacionamentos** [1].

Uma **entidade** é "algo" do mundo real com existência própria, podendo ter existência física (por exemplo, uma pessoa, um carro, uma casa) ou existência conceitual (por exemplo, uma empresa, um curso universitário). Um **relacionamento** é uma associação entre entidades, e cada entidade, exerce uma função num relacionamento, também chamado de papel, por exemplo: "Pai-Filho" é um relacionamento entre duas entidades "Pessoa", sendo uma entidade exercendo o papel de Pai e outra o papel de Filho. Tanto uma entidade quanto um relacionamento

podem ter propriedades particulares que os descrevem. Essas propriedades são denominadas de **atributos**. Os atributos de uma entidade ou de um relacionamento são obtidos por observação ou medida, por exemplo: uma entidade aluno pode ser descrita pelo nome, data de nascimento e endereço; um relacionamento entre as entidades universidade e aluno pode ser descrito pelo número e a data da matrícula [1, 5].

Diversas notações gráficas permitem visualizar os conceitos do modelo ER. Cada autor pode apresentar a representação gráfica para descrever os conceitos fundamentais do modelo ER. Demonstramos um exemplo desses conceitos (usando as notações gráficas de [5]) através do diagrama ER da Figura 2.

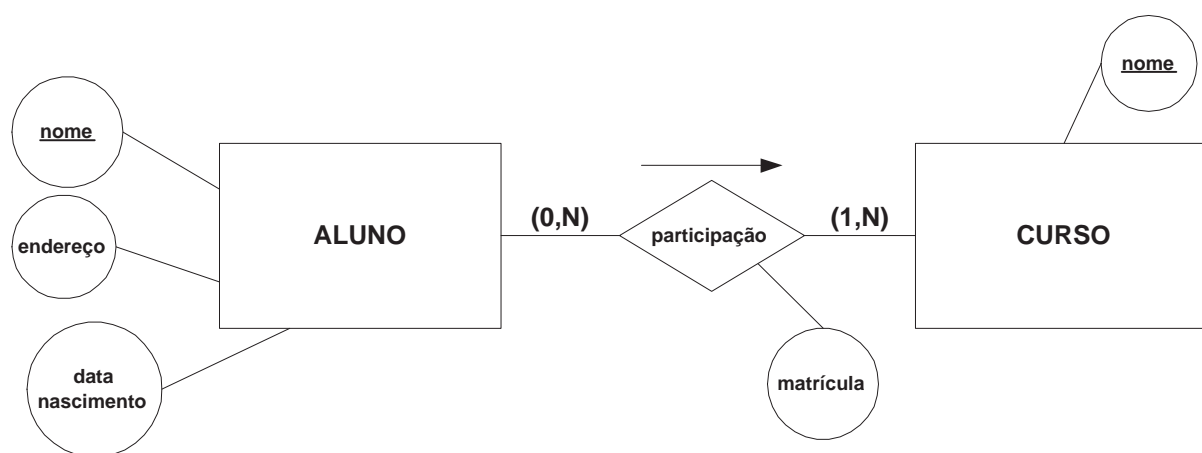


Figura 2: Exemplo de um Diagrama ER.

Na Figura 2 as entidades ALUNO e CURSO são representadas pelos retângulos, o relacionamento **participação** é representado pelo losango, e os atributos das entidades e dos relacionamentos são representados pelas elipses. Os atributos que estão sublinhados representam os atributos identificadores da entidade. Cada ocorrência de ALUNO está associada à, no mínimo, uma ocorrência de CURSO, e à, no máximo N, (onde N significa um número arbitrário de ocorrências) de CURSO. Já uma ocorrência de CURSO está associada à, no mínimo, zero ocorrências de ALUNO e à, no máximo N, ocorrências. Essa quantidade mínima e máxima de ocorrências de um relacionamento em que uma entidade pode participar é denominada de cardinalidade.

Os conceitos de modelagem conceitual ER descritos - entidade, relacionamento, atributo - podem modelar um conjunto de aplicações "tradicionais" de banco de dados, que incluem, geralmente, as aplicações de processamento de dados no comércio e na indústria, porém, aplicações mais complexas, como por exemplo, projetos de engenharia, sistemas de informações médicas, telecomunicações, etc., exigem conceitos adicionais para refletir com maior exatidão suas realidades [5].

Nesta seção, explicamos resumidamente o processo de projeto de banco de dados, ressaltamos o papel fundamental da modelagem conceitual dentro desse processo, e também, a técnica mais difundida para modelagem conceitual, que é o modelo ER. Na próxima seção apresentaremos como a Inteligência Artificial contribuiu com os conceitos mais avançados de modelagem conceitual, visando atender o processamento de informações das aplicações mais complexas.

3 Aplicações Práticas de Inteligência Artificial para Modelagem Conceitual de Banco de Dados

Desde o final dos anos 70, os projetistas de bancos de dados têm tentado projetar esquemas de bancos de dados mais exatos e completos que reflitam as propriedades e as restrições dos dados mais precisamente. Isso foi particularmente importante nas recentes aplicações da tecnologia de banco de dados, como os bancos de dados para projeto de engenharia e manufatura, telecomunicações, sistemas de informações geográficas, entre outras. Esses tipos de aplicações exigem requisitos mais complexos que a maioria das aplicações tradicionais. Objetivando atender as demandas crescentes de processamento de informações dessas aplicações complexas, tornou-se necessário o desenvolvimento de conceitos adicionais de modelagem semântica de dados, para serem incorporados aos modelos de dados conceituais, como o modelo ER. Muitos desses conceitos foram desenvolvidos na área de representação do conhecimento da Inteligência Artificial [5].

Na subseção a seguir, apresentamos algumas técnicas de representação do conhecimento que foram utilizadas para expandir os conceitos do modelo ER, visando aumentar o poder de expressão da modelagem conceitual de banco de dados.

3.1 Representação do Conhecimento

Um dos problemas complexos encontrados na Inteligência Artificial é a questão que trata do conteúdo que deve ser colocado na base de conhecimento de um agente, ou seja, como representar fatos sobre o mundo real, para que o agente possa compreendê-los e atuar de maneira inteligente. A área da Inteligência Artificial que estuda como representar ou formalizar esses fatos em um formato que seja tratável por computador é denominada de Representação do Conhecimento (abreviado por RC).

RC pode ser definida como "algo" que substitui o objeto ou fenômeno real, de modo a permitir a uma entidade determinar as conseqüências de um ato pelo pensamento ao invés de sua realização. Uma RC pode ser entendida como uma forma sistemática de estruturar e codificar o que se sabe sobre uma determinada aplicação. Portanto, ao contrário de uma codificação qualquer, uma RC deve apresentar as seguintes características [5, 8]:

- Ser compreensível ao ser humano.
- Abstrair-se dos detalhes de como funciona internamente o processador de conhecimento que a interpretará. RC faz uso de um processo de abstração para identificar as propriedades comuns e aspectos importantes do minimundo (domínio do discurso), enquanto suprimem as diferenças insignificantes e os detalhes sem importância.
- Ser robusta, isto é, permitir sua utilização mesmo que não aborde todas as situações possíveis.
- Ser generalizável, ao contrário do conhecimento em si que é individual. Uma representação necessita de vários pontos de vista do mesmo conhecimento, de modo que possa ser atribuída a diversas situações e interpretações.

Muitas técnicas de RC têm sido estudadas e exploradas pelos pesquisadores de Inteligência Artificial [9, 10]. O objetivo dessas técnicas é produzir conceitos para a modelagem mais exata possível de alguns domínios do conhecimento para a criação de uma ontologia que descreva os conceitos do domínio, e isso será usado para armazenar e manipular o conhecimento

para a geração de inferências, tomadas de decisões ou simplesmente para responder perguntas [5].

A seguir, abordaremos alguns conceitos e técnicas de RC que foram utilizadas na prática para aumentar os conceitos da modelagem conceitual de banco de dados, mais especificamente no modelo ER.

3.1.1 Redes Semânticas

Foram originalmente projetadas como um meio de representar os significados de palavras da língua inglesa. A informação é representada como um conjunto de nós conectados um ao outro por um conjunto de arcos rotulados que representam relações entre os nós [9].

Uma rede semântica também pode ser considerada como um grafo direcionado formado por um conjunto de nós que representam os objetos (por exemplo: indivíduos, coisas, conceitos, situações em um domínio) e por um conjunto de arcos que representam as relações entre os objetos. Um arco é rotulado com o nome da relação que ele representa, e vários arcos podem ter o mesmo rótulo, porém, cada objeto é representado por apenas um nó [8].

Na maioria das vezes, objetos complexos podem ser decompostos em objetos mais simples. Essas decomposições produzem dois tipos de relações [8]:

- classe-de (*is-a*): as relações entre os objetos estão em uma taxonomia hierárquica.
- faz-parte (*part-of*): as relações entre os objetos obedecem a um tipo de composição, ou seja, um objeto é componente de outro, não havendo nenhum tipo de herança.

Existem muitas variações de redes semânticas, mas todas são capazes de representar objetos individuais, categorias de objetos e relações entre objetos. Uma categoria é uma forma de organizar os objetos individuais. Pode-se também, imaginar uma categoria como o conjunto de seus elementos ou como um objeto mais complexo [10].

Uma parte de uma rede semântica típica apresentamos na Figura 3 [9].

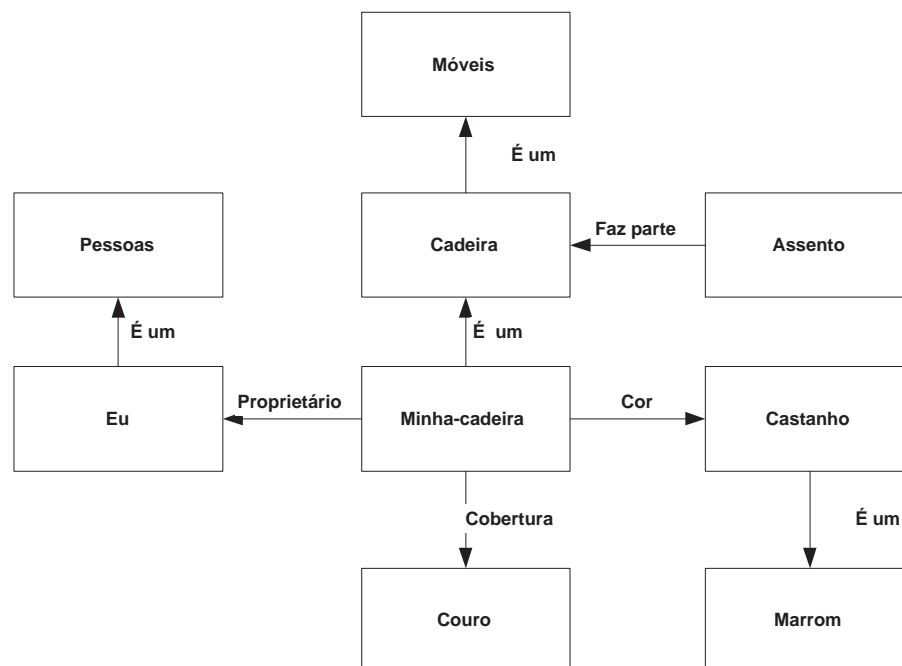


Figura 3: Exemplo de uma Rede Semântica.

Uma outra notação gráfica típica apresentamos na Figura 4, na qual exibe nomes de objetos ou categorias em elipses ou retângulos e os conecta por meio de arcos rotulados [10].

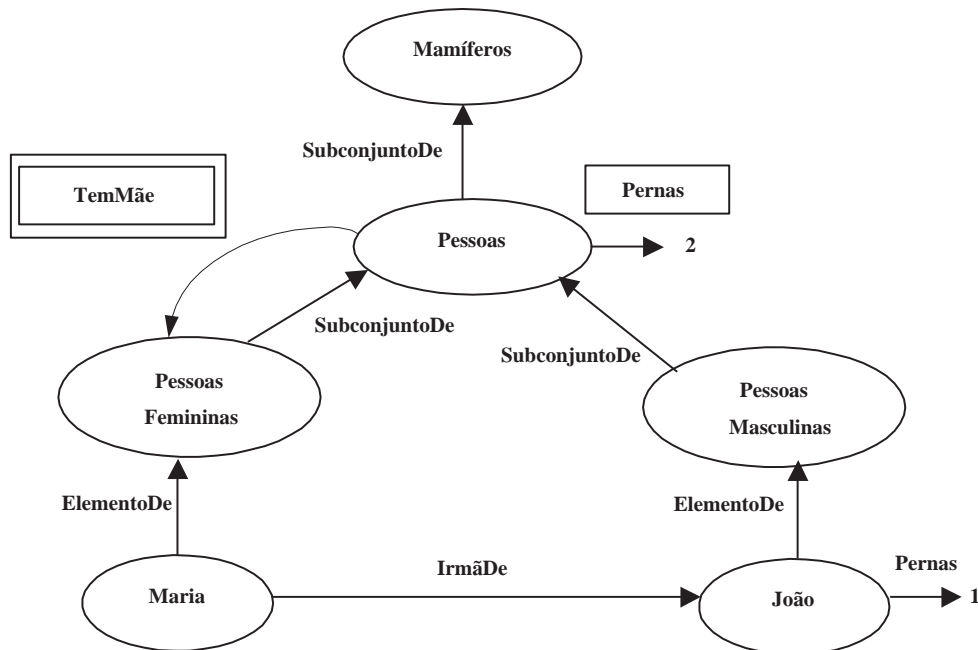


Figura 4: Uma Rede Semântica com quatro objetos (João, Maria, 1 e 2) e quatro categorias.

Na Figura 4 há um arco **ElementoDe** entre **Maria** e **PessoasFemininas**, que corresponde à asserção lógica **Maria** \in **PessoasFemininas**; de modo semelhante, o arco **IrmãDe** entre **Maria** e **João** corresponde à asserção **IrmãDe(Maria,João)**. Pode-se conectar as categorias usando arcos **SubconjuntoDe** e assim por diante. O arco identificado por retângulo de aresta dupla, rotulado como **TemMãe**, é uma relação entre uma pessoa e sua mãe. O arco de retângulo com arestas simples é usado para afirmar propriedades de todos os elementos de uma categoria.

É bastante útil visualizar uma rede semântica através de uma notação gráfica, como mostrado nas Figuras 3 ou 4, embora elas não possam ser representadas em um programa desse modo. Elas são representadas usando algum tipo de estrutura de memória de valor de atributo, como por exemplo, na linguagem de programação LISP, onde cada nó seria um átomo, os elos seriam propriedades e os nós, nas outras extremidades dos elos, seriam os valores. Vejamos no Código 1, como a rede semântica apresentada na Figura 3 seria representada em LISP [9].

Código 1 – Lista de Propriedades

- 1 ÁTOMO ((ÉUM MÓVEIS))
 - 2
 - 3 CADEIRA ((ÉUM CADEIRA)) ((COR CASTANHO))
 - 4
 - 5 MINHA-CADEIRA ((COBERTURA COURO)) ((PROPRIETÁRIO EU))
 - 6
 - 7 EU ((ÉUM PESSOA))
 - 8
 - 9 CASTANHO ((ÉUM MARROM))
 - 10
 - 11 ASSENTO ((FAZPARTEDE CADEIRA))
-

Uma das propriedades mais importantes das relações entre objetos numa rede semântica é a transitividade, pois permite uma declaração concisa de propriedades nos objetos mais gerais. Mecanismos de inferência podem ser usados para derivar essas propriedades para os objetos mais específicos. Esse procedimento é chamado **herança de propriedade** [8]. A herança é considerada uma maneira muito importante de inferência, porque permite que as propriedades de objetos sejam inferidas a partir de sua pertinência a categorias [10].

Como explicamos anteriormente, as categorias de objetos organizam os objetos individuais, além disso, também servem para organizar e simplificar a base de conhecimento por **herança**. Por exemplo: se afirmarmos que todas as instâncias da categoria **Alimento** são comestíveis e que **Fruta** é uma subclasse de **Alimento** e que **Maçãs** é uma subclasse de **Fruta**, então saberemos que toda maçã é comestível. Isto é, as maçãs individuais herdam a propriedade de serem comestíveis por serem elementos da categoria **Alimento** [10].

As relações de **subclasse** organizam categorias em uma **taxonomia** ou **hierarquia taxonômica**. As taxonomias foram utilizadas durante séculos em campos técnicos, como por exemplo: a biologia pretende fornecer uma taxonomia de todas as espécies vivas e extintas; as autoridades governamentais criaram taxonomias de ocupações e produtos comerciais; a ciência da informação desenvolveu uma taxonomia de todos os campos do conhecimento. Portanto, as taxonomias caracterizam um ponto importante do conhecimento comum em geral [10].

Analisando novamente a notação da rede semântica da Figura 4, podemos executar o raciocínio de herança, que acabamos de explicar, da seguinte forma: pelo fato de ser uma pessoa, **Maria** herda a propriedade de ter duas pernas. Para descobrir quantas pernas **Maria** tem, o algoritmo de herança segue o arco **ElementoDe** desde **Maria** até a categoria a que ela pertence, e depois segue o arco **SubconjuntoDe** pela hierarquia, até encontrar uma categoria para a qual exista um arco **Pernas** identificado por um retângulo, nesse caso, a categoria **Pessoas**. Esse mecanismo de inferência, por sua simplicidade e eficiência, foi um dos principais fatores de atração das redes semânticas [10].

A herança torna-se complexa quando um objeto pode pertencer a mais de uma categoria ou quando uma categoria pode ser um subconjunto de mais de uma outra categoria. Esse comportamento é denominado de **herança múltipla**. Quando isso ocorre, o algoritmo de herança pode encontrar dois ou mais valores conflitantes que respondem à consulta. Geralmente, isso é permitido em redes semânticas.

3.1.2 Disjunção

No item anterior, Redes Semânticas, informamos que as relações de **subclasse** organizam as categorias de forma hierárquica. Porém, também há necessidade de ter a possibilidade de expressar relações entre categorias que não são subclasses umas das outras, como por exemplo: se afirmássemos que **Machos** e **Fêmeas** são subclasses de **Animais**, não teríamos dito que o macho não pode ser uma fêmea. Portanto, para atender essa necessidade, é necessário caracterizar melhor as relações de subclasse, logo, é dito que duas ou mais categorias são **disjuntas** se elas não têm elementos (ou membros) em comum [10].

3.1.3 Composição Física

É habitual a idéia de que um objeto pode fazer parte de outro objeto, como por exemplo: a boca de uma pessoa faz parte da cabeça de uma pessoa, a Amazônia faz parte do Brasil, o carburador faz parte do carro, etc. Para representar esse tipo de "vínculo", é usado a relação geral **ParteDe** para dizer que algum objeto faz parte de outro. Os objetos podem ser agrupados em hierarquias de **ParteDe**, ou seja, como se fosse uma hierarquia de **Subconjunto**. Exemplo:

Ci₂¹igo 2 – Exemplo de Composição Física

- 1 ParteDe (Bucareste, Romênia)
 - 2 ParteDe (Romênia, Europa Oriental)
 - 3 ParteDe (Europa Oriental, Europa)
 - 4 ParteDe (Europa, Terra)
-

A relação **ParteDe** é transitiva e reflexiva, ou seja: **ParteDe(x,y) ∧ ParteDe(y,z) ⇒ ParteDe(x,z)**. Portanto, conclui-se que: **ParteDe(Bucareste,Terra)**, ou seja, que o objeto **Bucareste** faz parte do objeto **Terra** [10].

3.1.4 Lógicas Descritivas

As lógicas descritivas são notações projetadas para tornar mais fácil descrever definições e propriedades de categorias. Essas lógicas evoluíram a partir das redes semânticas. Assim, como as redes semânticas, as lógicas descritivas, foram elaboradas para auxiliar na organização de uma hierarquia de categorias. As principais tarefas de inferência para lógicas descritivas são a **subsunção** (conceito usado para verificar se uma categoria é um subconjunto de outra pela comparação de suas definições), e a **classificação** (conceito usado para verificar se um objeto pertence a uma categoria). Uma lógica descritiva típica é a linguagem CLASSIC. Vejamos um exemplo de sua sintaxe [10]: **Solteiro = And (NãoCasado, Adulto, Homem)** O equivalente em lógica de primeira ordem seria: **Solteiro(x) ⇔ NãoCasado(x) ∧ Adulto(x) ∧ Homem(x)**.

3.1.5 Frames

Frames (ou Quadros) é uma teoria criada por Minsky, em 1975. Esse termo é utilizado para denominar um grupo de conhecimentos relevantes a uma determinada "coisa", um indivíduo, uma situação (por exemplo: entrar em uma sala de estar ou descrever uma determinada cena) ou um conceito. O *frame* possui um nome que identifica o conceito por ele definido e consiste de uma coleção de atributos, denominados *slots*. Um *frame* é uma estrutura de dados complexa, poderosa e bastante expressiva, porém, oferece uma maneira útil de modelar objetos do mundo real [8].

Todo *frame* possui um nome que o identifica e referencia, detalhes de seus *frames*-pais e um conjunto de *slots* que contêm valores ou ponteiros para valores. Um *slot* possui um nome e é formado de um conjunto de atributos com informações que o descreve. Esse conjunto é chamado de *facetas*. As informações contidas nas *facetas* definem de maneira explícita os valores que o *slot* pode assumir, ou pode indicar a forma de calcular ou inferir o seu valor. Vejamos alguns exemplos de *facetas*: tipo, domínio, valor *default*. etc. Os valores dos *slots* podem ser definidos de forma explícita ou herdados de modo implícito por meio de um de seus ancestrais [8].

Herança de propriedades é considerada uma das principais características existentes em *frames*, na qual uma classe mais específica pode herdar todas as propriedades da classe mais genérica. As associações entre *frames* determinam a sua estrutura hierárquica. Toda associação liga um *frame*-pai ao seu filho. O *frame*-filho pode ser compreendido como uma **especialização** do *frame*-pai, ou o *frame*-pai como uma **generalização** do *frame*-filho. Isso também significa que, um *frame*-filho pode herdar valores (*default* ou correntes) de qualquer um dos seus *frames*-pais, que por sua vez também herdam de seus pais, e assim por diante. Dessa forma, a informação é distribuída sem duplicação [8].

Dentro da hierarquia de *frames*, as instâncias aparecem como nós-folhas, e elas podem conter somente um pai e apenas valores correntes em seus *slots*. Geralmente, a informação navega partindo dos *frames* do topo para os das extremidades. Esse comportamento é realizado pela herança. Quando houver uma solicitação de algum valor de um *slot*, o algoritmo de herança é automaticamente invocado. O local mais provável para procurar o valor é no próprio *frame* original, visto que ele pode ter o valor corrente e/ou o valor *default* para o atributo requisitado [8].

Através da hierarquia de *frames* é permitido que os dados sejam guardados de maneira abstrata e aninhada com propriedades comuns que são automaticamente herdadas por meio da hierarquia. A estrutura de *frames* evita a duplicação desnecessária de informações, além do ganho de algumas vantagens como: simplificação do código implementado e facilidades de leitura e manutenção [8].

Na Figura 5 fornecemos uma idéia gráfica do modelo de representação de conhecimento baseado em *frames* [3].

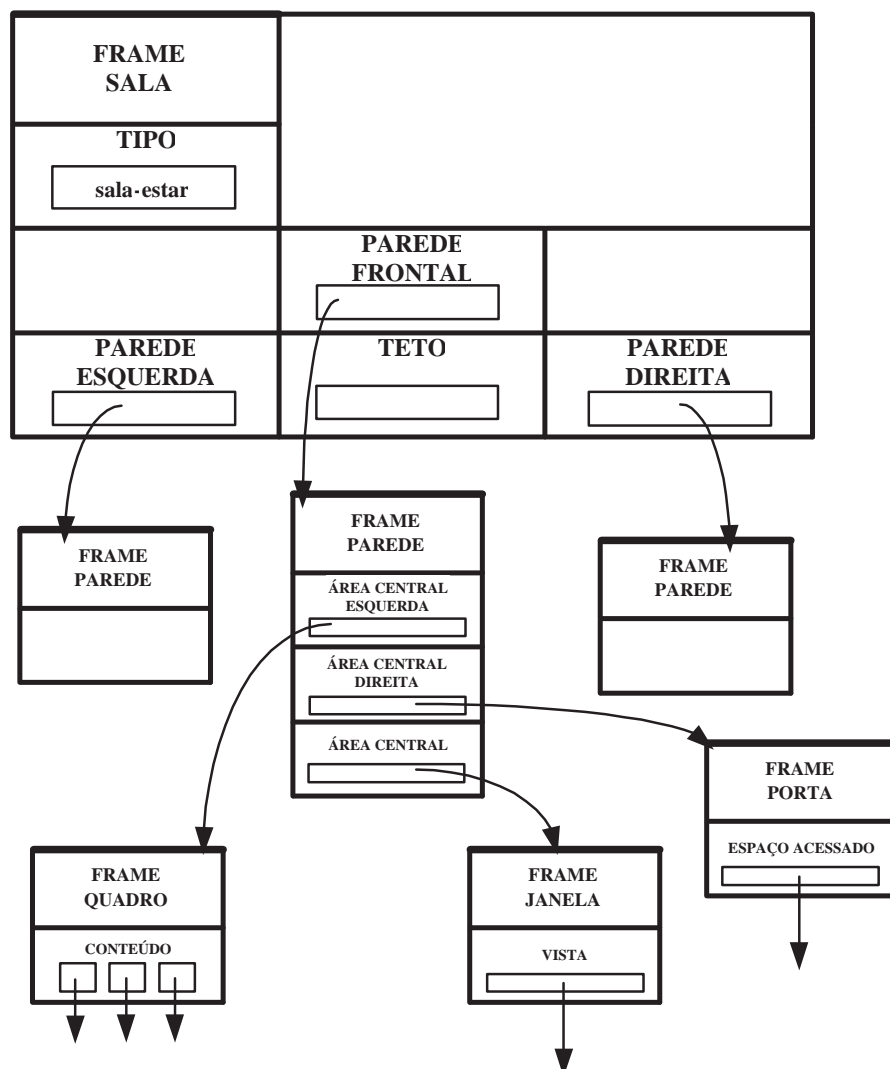


Figura 5: Representação de uma sala usando *Frames*.

No exemplo apresentado na Figura 5, as lacunas são preenchidas por valores e ponteiros para outros *frames*. Quando um *frame* é requisitado para utilização, algumas de suas lacunas

podem já estar preenchidas, são valores *default*. Estas lacunas, como as outras serão ou não completadas de acordo com a situação que se apresenta. Este fato pode ser compreendido se supusermos o seguinte fato: ao entrarmos em um sala, imaginamos logo um paralelepípedo com portas e janelas nas paredes. Estes são os dados padrão, e os detalhes a respeito do tapete, dos móveis, adornos, etc. são completados com os dados necessários para descrever corretamente a sala. Isto é o que está indicado na Figura 5 [3].

3.1.6 Ontologias

No início dessa seção informamos que um dos objetivos das técnicas de RC é produzir conceitos para a modelagem mais exata possível de alguns domínios do conhecimento, visando a formação de uma ontologia. O termo empregado, **ontologia**, está intimamente ligado à RC, por isso, o incluímos nessa Subseção (3.1). Ele tem origem na filosofia e na metafísica. A definição mais usada de ontologia é "a especificação de uma conceitualização", isto é, uma **conceitualização** é o conjunto de conceitos que são usados para representar a parte da realidade ou conhecimento que é importante para uma comunidade, e **especificação** é referente à linguagem e aos termos de vocabulário que são usados para especificar a conceitualização. Os estudos que envolvem ontologias procuram descrever as estruturas e os relacionamentos que são possíveis na realidade através de algum vocabulário comum [5].

Algumas técnicas possíveis visam descrever as ontologias, são elas [5]:

- **Thesaurus**: é um dicionário de termos que descreve os relacionamentos entre as palavras que representam diversos conceitos.
- **Taxonomia**: uma taxonomia que descreve como os conceitos de uma determinada área de conhecimento são relacionados utilizando estruturas semelhantes àquelas usadas em uma especialização ou generalização.
- **Esquema Conceitual de Banco de Dados**: alguns consideram um esquema conceitual de banco de dados detalhado como uma ontologia, que descreve os conceitos (entidades e atributos) e relacionamentos de um pequeno mundo real.
- **Teoria Lógica**: uma teoria lógica faz uso dos conceitos de lógica matemática para definir conceitos e seus relacionamentos.

Geralmente, os conceitos usados para descrever as ontologias são muito parecidos aos que explicamos na Seção 2, como entidades, atributos e relacionamentos. A grande diferença entre uma ontologia e um esquema conceitual de banco de dados, é que o esquema limita-se a descrever um pequeno subconjunto de um minimundo real, para armazenar e gerenciar dados, enquanto que uma ontologia, é considerada mais genérica e completa possível naquilo que ela se propõe em descrever, ou seja, uma parte da realidade.

Nessa subseção abordamos algumas técnicas e conceitos de RC cujos conceitos abstratos foram aproveitados de maneira concreta na modelagem conceitual de banco de dados, especificamente no modelo ER, levando ao modelo ER estendido ou modelo EER. Os detalhes desse aproveitamento serão explicados na próxima subseção.

3.2 Abstração de Dados

Em [6] temos que o termo abstração é o "ato de separar um ou mais elementos de uma totalidade complexa (coisa, representação, fato), os quais só mentalmente podem subsistir fora

dessa totalidade". Em outras palavras, abstrair significa isolar, seletivamente, características relevantes do todo observado, desconsiderando detalhes adicionais que, de outro modo, conturbariam o processo cognitivo.

Uma abstração permite capturar um determinado aspecto existente em uma realidade, eliminando detalhes irrelevantes. Essa técnica é bastante usada na criação de um *software*.

A seguir, apresentamos os tipos de conceitos de representação de dados abstratos que são amplamente usados na modelagem conceitual de banco de dados:

3.2.1 Classificação e Instanciação

O processo de **classificação** abrange, sistematicamente, a correlação de mesma natureza entre os objetos/entidades com as classes de objeto/tipos entidade. Assim, pode-se descrever (em banco de dados) ou raciocinar (em RC) a respeito de classes em vez de objetos individuais. Os conjuntos de objetos compartilham os mesmos tipos de atributos, relacionamentos e restrições, e pela classificação de objetos simplifica-se o processo de descoberta de suas propriedades comuns [5]. Todas as características que não são compartilhadas por todos são abstraídas. Exemplificando: a classe "Jornal" define propriedades comuns aos objetos "Correio Brasiliense", "Folha de São Paulo", "O Popular". Através do exemplo dado, podemos verificar que de um grupo heterogêneo busca-se encontrar um conceito comum.

A abstração classificação faz uso das idéias de uma das principais tarefas de inferência, também denominada classificação, usada para lógicas descritivas (técnica de RC).

Instanciação é o processo inverso da classificação e se refere à criação e ao exame específico dos diferentes objetos de uma classe. Essas instâncias geradas são objetos e possuem todas as propriedades de sua classe. Exemplo: "Folha de São Paulo" é uma instância da classe "Jornal". Uma instância é o resultado da aplicação da abstração de instanciação sobre uma classe de objetos. Portanto, uma instância de objeto está relacionada à sua classe de objeto pelo relacionamento **E-UMA-INSTANCIA-DE** ou **E-UM-MEMBRO-DE**. No modelo EER, as entidades são classificadas em tipos entidade, de acordo com seus atributos e relacionamentos básicos e as instâncias de relacionamento são classificadas em tipos relacionamento [5].

3.2.2 Identificação

Identificação é o processo de abstração pelo qual as classes e os objetos são identificados unicamente por meio de algum **identificador**, como por exemplo: um nome de classe identifica univocamente toda uma classe. Porém, um mecanismo adicional é preciso para distinguir as instâncias de objetos, através do significado dos identificadores de objeto. Além disso, também é preciso identificar as várias manifestações no banco de dados do mesmo objeto do mundo real. Exemplificando: podemos ter a tupla <Geoflória, 2005, 288-5555> em uma relação **PESSOA**, e outra tupla <301-5265, XX, 3.99> em uma relação **ALUNO**, que ocorrem para representar a mesma entidade do mundo real. Não há como identificar o fato de que essas duas tuplas representam a mesma entidade do mundo real, a não ser que façamos o registro, em tempo de projeto, da referência cruzada adequada que forneça essa identificação. Portanto, o processo de identificação é preciso em dois níveis: 1) para a diferenciação entre os objetos e classes do banco de dados; 2) para identificar os objetos do banco de dados e relacioná-los aos seus correspondentes no mundo real [5].

No modelo EER, a identificação dos construtores de esquema é baseada em um sistema de nomes únicos para essas construções. Como por exemplo, toda classe em um esquema EER deve ter um nome distinto, e os nomes dos seus atributos também devem ser distintos [5].

3.2.3 Especialização e Generalização

Especialização é o processo que permite o refinamento de um conceito (ou objeto). O resultado desse refinamento origina um conjunto de **subclasses**, e essas subclasses devem ser compatíveis com o conceito mais genérico de um tipo entidade; esse tipo entidade é chamado **superclasse** da especialização [5]. Por exemplo, o conjunto de subclasses **ANALISTA DE SISTEMAS, PROGRAMADOR, DIGITADOR** é uma especialização da superclasse **PROFISSIONAL DE INFORMÁTICA**, que distingue cada entidade profissional com base em seu tipo de trabalho. A Figura 6 mostra uma forma de representar diagramaticamente uma especialização em um diagrama EER.

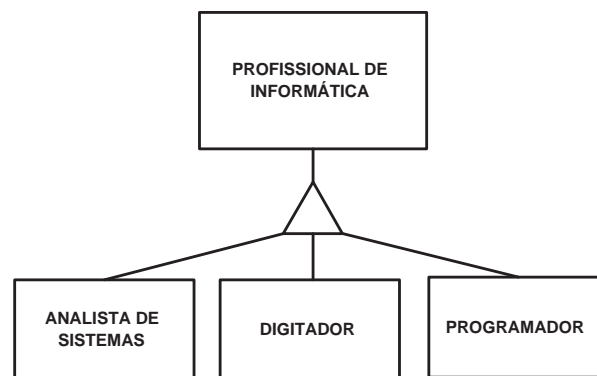


Figura 6: Notação de diagrama EER para representar as subclasses e a especialização.

Duas razões levam à inclusão de relacionamentos classe/subclasse e especializações em um modelo de dados conceitual, vejamos: a primeira é que alguns atributos podem ser usados em algumas, mas não em todas as entidades da superclasse. Uma subclasse consiste em agrupar as entidades para as quais esses atributos se aplicam. Os membros da subclasse podem compartilhar a maioria de seus atributos com outros membros da superclasse; a segunda razão é que somente as entidades que sejam membros de alguma subclasse podem participar de algum tipo de relacionamento [5].

Em síntese, o processo de especialização permite [5]:

- Definir uma coleção de subclasses de um tipo entidade.
- Determinar atributos específicos adicionais para cada subclasse.
- Determinar tipos relacionamentos adicionais específicos entre cada subclasse e outros tipos entidade, ou outras subclasses.

Generalização é o processo inverso da especialização (ou é o dual da especialização). Esse processo permite abstrair ou ocultar as diferenças entre classes de objetos e as propriedades comuns são resumidas em um único tipo entidade, resultando uma **superclasse**. O termo generalização é usado para referir ao processo de definição de um tipo entidade generalizada a partir de tipos entidade que foram dados [5].

O relacionamento entre uma subclasse e sua superclasse é chamado relacionamento E-UMA-SUBCLASSE-DE ou, simplesmente, um relacionamento IS-A (ou E-UM). Conforme exemplo demonstrado na Figura 6, dizemos "um Analista de Sistemas é um Profissional de Informática", "um Programador é um Profissional de Informática", e assim por diante.

A idéia de herança de propriedade, mecanismo de inferência utilizado em redes semânticas (técnica de RC), está totalmente associada ao conceito de generalização/especialização da modelagem conceitual de banco de dados.

Herdar propriedades significa que cada ocorrência da entidade especializada possui, além de suas próprias propriedades (atributos, relacionamentos e generalizações/especializações), também as propriedades da ocorrência da entidade genérica correspondente. Segundo demonstramos na Figura 7, a entidade **PESSOA FÍSICA** possui, além de seus atributos particulares, CPF e sexo, também todas as propriedades da ocorrência da entidade **CLIENTE** correspondente, ou seja, os atributos nome e código, o seu identificador (atributo código), assim como o relacionamento com a entidade **FILIAL** [7].

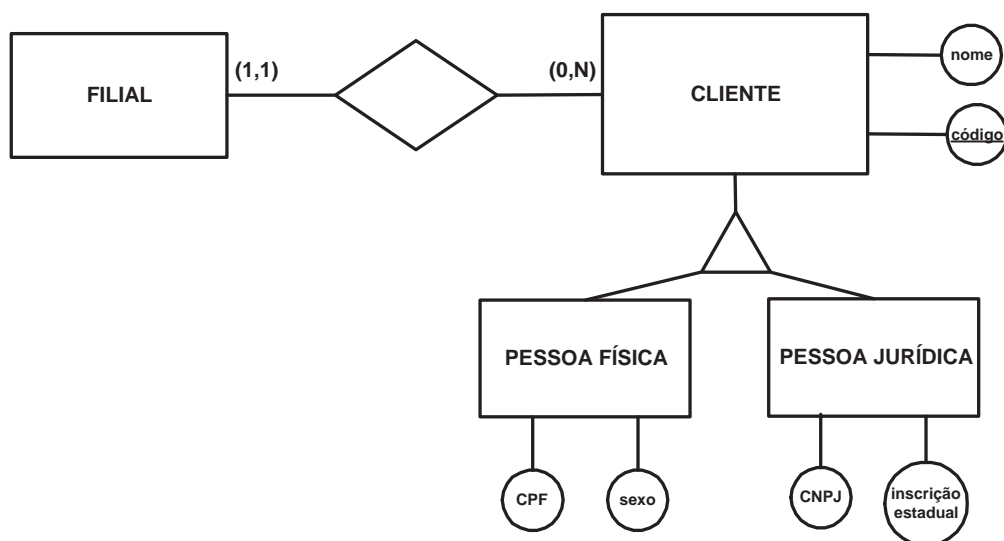


Figura 7: Notação de diagrama EER para representar a abstração generalização/especialização.

Em síntese, o diagrama expressa que toda pessoa física possui os atributos nome, código, CPF e sexo, e é identificada pelo código e está obrigatoriamente relacionada a uma filial. Da mesma forma, toda pessoa jurídica possui os atributos nome, código, CNPJ e inscrição estadual, e é identificada pelo código e está obrigatoriamente relacionada a uma filial [7].

Algumas **restrições** podem ser definidas no uso de especialização/generalização, vejamos [12, 5]:

- **Disjunção:** cada elemento da entidade genérica é mapeado para, no máximo, um elemento das entidades especializadas. Especifica que as subclasses da especialização devem ser mutuamente exclusivas.
- **Sobreposição:** ocorre quando um elemento da entidade genérica é mapeado para elementos de duas ou mais entidades especializadas. Usado quando as subclasses não estão condicionadas a serem disjuntas, portanto, seus conjuntos de entidades podem sobrepor-se.
- **Total:** cada elemento de uma entidade genérica é mapeado para pelo menos um elemento das entidades especializadas.
- **Parcial:** quando um elemento da entidade genérica pode não ter nenhum elemento correspondente nas entidades especializadas.

As restrições de integralidade (total e parcial) e de disjunção (disjunção e sobreposição) são independentes. Em função disso, é possível a seguinte combinação entre elas: disjunção total, disjunção parcial, sobreposição total e sobreposição parcial [5].

A característica de duas ou mais categorias serem **disjuntas**, conceito aproveitado de RC, é também aplicado para caracterizar melhor as relações de subclasse, dando origem as restrições de disjunção.

Uma dada subclasse pode conter suas próprias subclasses, formando uma **hierarquia ou um reticulado de especializações**. Exemplificando: na Figura 8, **VEÍCULO TERRESTRE** é uma subclasse de **VEÍCULO**, e é também uma superclasse de **VEÍCULO ANFÍBIO**; esta é uma restrição que certamente foi imposta pelo mundo real, que o veículo anfíbio deva ser um veículo terrestre [5, 7].

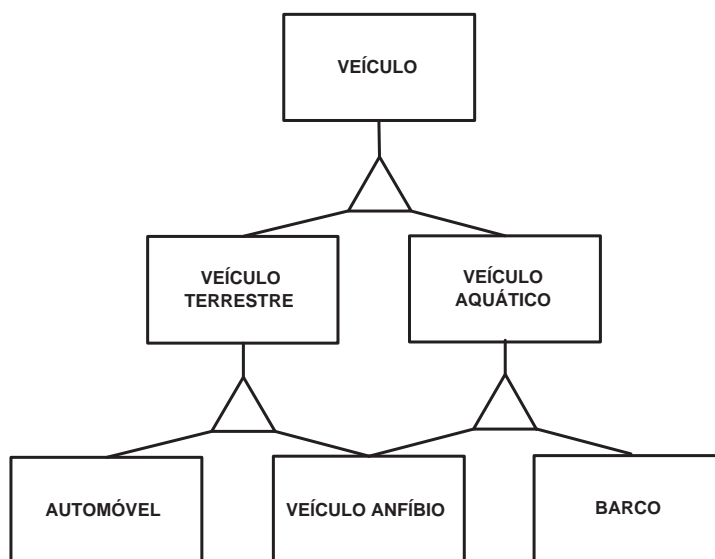


Figura 8: Um reticulado de especialização.

Uma **hierarquia de especialização** restringe todas as suas subclasses a participar de apenas um relacionamento classe/subclasse, ou seja, cada subclasse tem apenas uma classe genérica ou superclasse. Enquanto que na **especialização reticulada**, uma subclasse pode participar de mais de um relacionamento classe/subclasse, logo, a Figura 8 é um reticulado [5].

Em uma especialização reticulada ou hierárquica, uma subclasse herda as propriedades tanto da sua superclasse direta, mas também de todas as superclasses predecessoras, até o topo da hierarquia ou reticulado [5].

Uma subclasse com mais de uma superclasse é denominada uma subclasse compartilhada, como **VEÍCULO ANFÍBIO** na Figura 8. Esse tipo de comportamento em que a subclasse **VEÍCULO ANFÍBIO** herda, diretamente, as propriedades de múltiplas classes, é chamado por **herança múltipla**.

A Figura 8 apresenta um diagrama EER em que aparecem múltiplos níveis de generalização/especialização, assim como, o conceito de herança múltipla. Exemplificando: uma entidade **BARCO** é uma especialização de um **VEÍCULO AQUÁTICO**, que por sua vez é uma especialização de **VEÍCULO**. Assim um barco tem, além de suas propriedades específicas, também as propriedades de um veículo aquático e de um veículo em geral. O exemplo de herança múltipla aparece na entidade **VEÍCULO ANFÍBIO**. Um veículo anfíbio possui as propriedades de um veículo aquático, de um veículo terrestre e também suas propriedades específicas [7].

Destamos que a especialização reticulada utiliza o conceito de herança múltipla, que também é característico em redes semânticas; e as hierarquias de especialização fazem um forte uso dos conceitos de RC baseado em *frames* [5].

3.2.4 Agregação e Associação

Agregação é um conceito de abstração que permite a construção de objetos compostos a partir de objetos componentes. Vejamos alguns casos em que esse conceito pode ser relacionado ao modelo EER [5]:

- Refere-se à situação na qual agregamos os valores do atributo de um objeto para formar o objeto como um todo.
- É quando representamos um relacionamento de agregação como um relacionamento comum.
- Corresponde à possibilidade de combinar os objetos que estão relacionados por uma instância de relacionamento em particular em um objeto agregado de alto nível, esse caso não é fornecido de maneira explícita no modelo EER.

O relacionamento entre os objetos primitivos e seu objeto agregado é denominado E-UMA-PARTE-DE; o contrário é chamado de E-UM-COMPONENTE-DE [5]. Em outras palavras, podemos dizer que a agregação é uma forma especializada de "associação" na qual um objeto todo é relacionado com seu(s) objeto(s) parte, com isso, estabelece-se uma relação do tipo **Parte-Todo** entre conceitos. Nessa associação existe obrigatoriamente dependência existencial das "Partes" com o "Todo". Os objetos componentes ("Parte") são fortemente coesos, sendo assim, quando se aplica uma operação qualquer no "Todo", os objetos componentes também sofrerão essa operação.

A abstração **associação** estabelece uma associação entre objetos. Permite modelar uma conexão física ou conceitual entre objetos. É utilizada para associar os objetos de diversas classes independentes [5]. É representada no modelo EER por tipos relacionamentos, conhecidos como ESTA-ASSOCIADO-COM. Exemplificando: o objeto "O Popular", da classe "Jornal", está associado ao objeto "Gráfica Jaime Câmara", da classe "Editora".

Para melhor entendimento do conceito de agregação no modelo ER, explicaremos um de seus usos conforme demonstrado na Figura 9. Nesse exemplo temos as entidades **ALUNO**, **DEPARTAMENTO** e **CARTEIRA de BIBLIOTECA**, e um relacionamento "pertence a". Os alunos podem ou não pertencer a um departamento, e os departamentos podem conter alunos. Os alunos de qualquer departamento podem solicitar uma carteira da biblioteca, mas alunos sem departamento não podem. Assim, nem todos alunos possuem a carteira da biblioteca, e desta forma o modelo representado pela Figura 9 não representa a realidade requisitada [12].

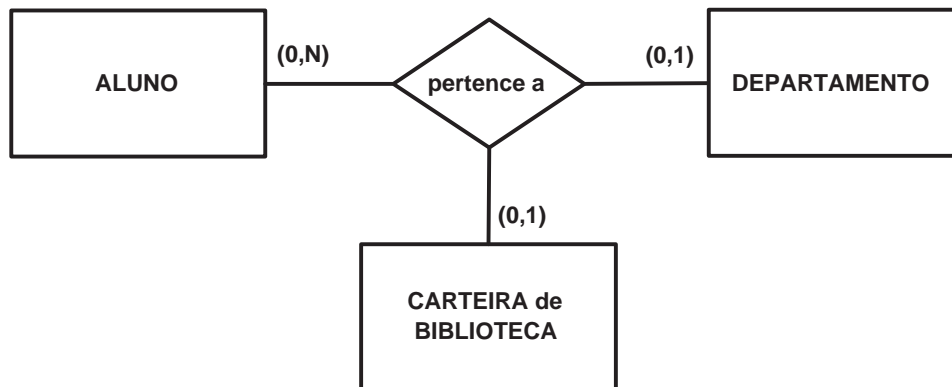


Figura 9: Diagrama EER inadequado conforme requisitos.

Uma alternativa para solucionar o problema é criar um novo relacionamento, chamado "emissão" e associá-lo ao relacionamento "pertence a". Mas, relacionar um relacionamento com outro relacionamento não é permitido no modelo ER. Portanto, o uso do conceito de agregação faz desse relacionamento uma entidade abstrata (ou classe agregada de alto nível), podendo assim, ser relacionada com uma outra entidade através de um novo relacionamento. A Figura 10 ilustra a nova solução. Lembrando que para cada ocorrência do par das entidades **ALUNO** e **DEPARTAMENTO** existe uma ocorrência da entidade **CARTEIRA de BIBLIOTECA**, caso o aluno a tenha requisitado [12].

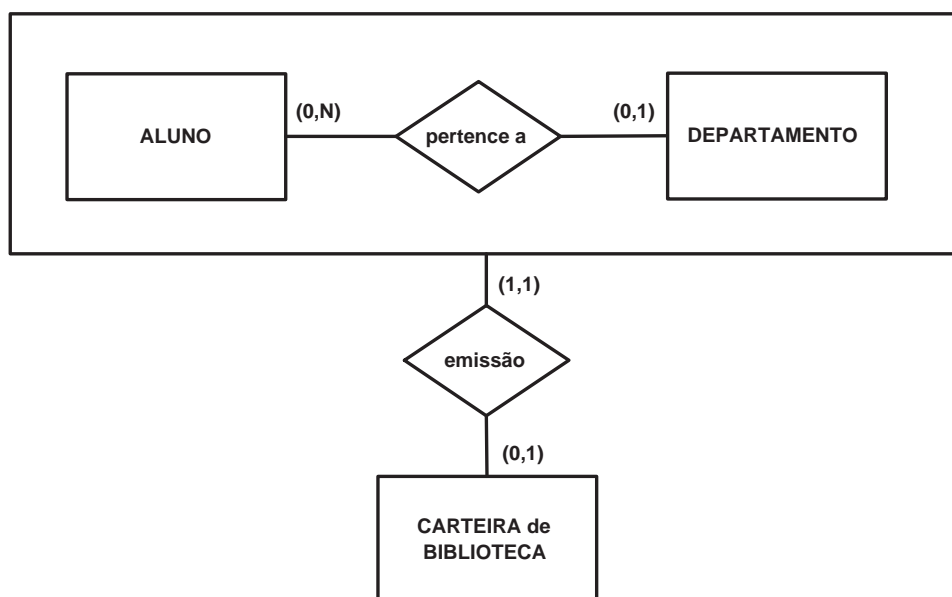


Figura 10: Representação gráfica de agregação.

A principal diferença estrutural entre agregação e associação é quando uma instância de associação for excluída, os objetos participantes podem continuar existindo. No entanto, para um objeto agregado, como por exemplo, um **CARRO**, que é composto pelos objetos **MOTOR**, **CHASSI** e **PNEUS**, se excluir o objeto agregado **CARRO** implica excluir todos os seus objetos componentes.

A idéia desse "vínculo" **ParteDe**, dado pelo conceito de composição física e Redes Semânticas, tem grande influência na origem do tipo de abstração agregação.

Nesta seção apresentamos alguns conceitos e técnicas de RC que foram usadas para aumentar o poder de expressão do modelo ER, e quais foram os novos tipos de conceitos de representação de dados abstratos que foram incorporados, dando origem ao modelo EER.

4 Considerações Finais

Para atender à exigência do processamento de dados dos sistemas de informações, as estruturas que um SGBD irá suportar precisa estar de acordo com as necessidades das aplicações do negócio dos usuários interessados. Portanto, ratificamos, como o papel da modelagem conceitual dentro de um processo de projeto de banco de dados é essencial, porque as estruturas e restrições de um banco de dados devem representar fielmente o mundo real em questão.

Além de modelar as estruturas de um banco de dados, a modelagem conceitual também se propõe a modelar processos e suas interações dentro do contexto de uma organização, ou seja, pode ser usada como ferramenta na gestão da organização, porque dá a possibilidade de se ter uma visão formal das informações, que pode ser entendida por pessoas leigas em computação e em formalismos matemáticos [11].

O modelo ER destaca-se como uma das técnicas mais usadas na modelagem conceitual de banco de dados. Porém, algumas aplicações complexas, exigiram a expansão dos seus construtores. Em busca de aumentar o poder de expressão do modelo ER, estudiosos em modelagem conceitual, buscaram em uma das áreas de estudo da Inteligência Artificial, que é a representação do conhecimento, conceitos e técnicas. Nesse relatório apresentamos uma pequena variedade de como se representa o conhecimento (redes semânticas, *frames*, etc), e quais foram as contribuições práticas que favoreceram as extensões para o modelo ER que melhoram sua capacidade de representação. Em [4] foi proposto um sistema para desenvolver esquemas conceituais de banco de dados baseados em *frames*.

5 Agradecimento

Ao Prof. Dr. Cedric Luiz de Carvalho pela avaliação do presente texto e pelas sugestões feitas, as quais muito contribuíram para a melhoria do texto original.

Referências

- [1] CHEN, P. P.-S. **The Entity-Relationship Model - Toward a Unified View of Data**. ACM Transactions on Database Systems, 1(1), 1976.
- [2] CHEN, P. P.-S. **The Entity-Relationship Model - A Basis for the Enterprise View of Data**. National Computer Conference, 1977.
- [3] DA CUNHA E SOUSA RIBEIRO, H. **Introdução aos Sistemas Especialistas**. LTC-Livros Técnicos e Científicos Editora, 1987.
- [4] DOS SANTOS, J. L. C; MEDEIROS, J. H. N; CATUNDA, R. H. C; SCHIEL, U. **DESCONT - Um Sistema para Desenvolvimento do Esquema Conceitual Baseado em Frames**. 3o. SBBD, 1988.
- [5] ELMASRI, R; NAVATHE, S. B. **Sistemas de Banco de Dados**. Addison Wesley, 4a. edition, 2005.

- [6] FERREIRA, A. B. H. **Novo Dicionário Aurélio da Língua Portuguesa**. Editora Nova Fronteira, 1986.
- [7] HEUSER, C. A. **Projeto de Banco de Dados**. Editora Sagra Luzzato, 4a. edition, 2001.
- [8] REZENDE, S. O. **Sistemas Inteligentes - Fundamentos e Aplicações**. Editora Manole, 2003.
- [9] RICH, E. **Inteligência Artificial**. Editora McGraw-Hill, 1988.
- [10] RUSSEL, S. J; NORVIG, P. **Artificial Intelligence. A Modern Approach**. Prentice-Hall, 2a. edition, 2003.
- [11] SETZER, V. W. **Projeto Lógico e Projeto Físico de Bancos de Dados**. Editora Gráfica Formato Ltda, 1986.
- [12] WOLPP, R. L. G. **Modelagem de uma Ferramenta para Construção de Esquema de Banco de Dados usando o MER**. Monografia do curso Especialização em Bancos de Dados do Instituto de Informática da Universidade Federal de Goiás, 2004.