6 Frames

6.1 Introdução

Um sistema computacional, da mesma forma que um sistema de conhecimento, constitui um modelo do mundo real, sobre o qual ele contém informação específica pertinente a alguma aplicação particular. Baseado nessa afirmação, o processo de especificação de software pode ser visto como um processo de construir um modelo apurado de algum empreendimento. Esta visão introduz o foco do nível de engenharia. Neste nível, um sistema gerenciador de bases de conhecimento (SGBC) é visto do ponto de vista do projetista da base de conhecimento, isto é, do engenheiro de conhecimento, o qual considera o SGBC em termos de como o conhecimento sobre uma aplicação pode ser modelado.

Portanto, o ponto inicial no projeto deste nível é a escolha do modelo (quadro de referência conceitual) de representação de conhecimento a ser fornecido ao engenheiro de conhecimento na interface do nível (interface centrada em objetos). Este sistema de conhecimento deve permitir uma representação acurada da visão de mundo do usuário. Para projetar tal sistema de conhecimento é necessário identificar as construções que as pessoas aplicam durante o processo de descrever o mundo real.

Isto parece assinalar o trabalho sendo feito em modelagem de dados na área de sistemas de bancos de dados bem como em representação de conhecimento na área de inteligência artificial. Os dois campos comportam a observação feita acima, buscando, por essa razão, o desenvolvimento de modelos que também reflitam tal conceitualização natural do usuário. A maior diferença entre o modelo de conhecimento do nível de engenharia e estes outros esquemas de representação reside na ênfase que eles dão aos diferentes tipos de conhecimento a ser modelado. Ao invés de ser declarativo, procedural ou estrutural, com respeito à ênfase em tais tipos, o modelo de conhecimento deve permanecer neutro, focalizando igualmente a todos. Ele deve poder fundir vários conceitos através da integração de todos os diferentes aspectos do conhecimento com o fim de obter o necessário quadro de referência poderoso, para permitir uma modelagem natural de todos os aspectos de cada mundo de aplicação.

Existem basicamente três tipos de conhecimento envolvidos em uma aplicação de sistema de conhecimento. O primeiro tipo, isto é, o tipo declarativo, refere-se ao conhecimento sobre partes passivas do domínio da aplicação. Considerado por quase todos os modelos de dados e esquemas de representação de conhecimento, ele define informações sobre os objetos existentes, seus atributos, restrições importantes, etc. O segundo tipo define as características procedurais do domínio da aplicação, bem como as estratégias humanas na resolução de problemas neste domínio, e é assim chamado de conhecimento procedural. Finalmente, o terceiro tipo, isto é, o conhecimento estrutural define a maneira de organizar os outros dois. Claramente, estes três tipos de conhecimento devem ser refletidos em construções diferentes do modelo de conhecimento. Eles definem três diferentes aspectos dele, que serão chamados de descritivo, operacional e organizacional.

Neste texto, estes diferentes aspectos de um modelo de conhecimento serão discutidos de um ponto de vista neutro: eles não são considerados a partir de uma realização em um sistema de representação de conhecimento particular, nem de um modelo de dados, mas da conceitualização do mundo real, visto por observadores humanos neste contexto, por engenheiros de conhecimento.

6.2 Aspectos Descritivos

Os aspectos descritivos do modelo de conhecimento tratam das características declarativas do mundo da aplicação. Por esta razão eles são convenientes para descrever as partes passivas do domínio de conhecimento, devido a sua natureza inativa.

Outras abordagens que têm o mesmo propósito são, por exemplo, modelos de dados. De maneira geral, eles são baseados na noção de "registro" (com campos numéricos ou alfanuméricos) e algumas "ligações" entre eles. Entretanto, estas primitivas são mais apropriadas para modelar o modo através do qual os dados são armazenados e acessados no computador do que para representar os conceitos do mundo real que estão por trás desses dados e seus relacionamentos.

Ao contrário dos modelos de dados, o modelo de conhecimento fornecido pelo nível de engenharia não deve necessariamente representar o mundo de uma forma ótima a nível de programação, mas refletir exatamente a aparência deste mundo.

Assim, o nível de engenharia vê as partes declarativas do mundo de aplicação compreendidas em termos de *objetos* conceituais, também chamados de entidades, que têm descrições (isto é, atributos ou *propriedades*) associados a eles, e têm também *relacionamentos* entre si de forma significante. Ademais, certas atividades (isto é, *operações* sobre estes objetos) ocorrem com o tempo, resultando em mudanças nos objetos e em seus inter-relacionamentos. Tanto o objeto (incluindo propriedades e relacionamentos) quanto as descrições de atividades estão sujeitos a *restrições* que definem os conceitos e distinguem a "realidade" de outros mundos possíveis.

O termo objeto é usado aqui para referir a uma estrutura de dados na base de conhecimento que pretende denotar uma entidade do mundo real. Por exemplo, uma base de conhecimento poderia conter um objeto para representar a pessoa Maria. Objetos são também usados para representar conceitos abstratos tais como pessoa, carro, etc.

Esta observação provê a motivação para o mais importante axioma do modelo de conhecimento: *tudo o que existe no mundo real, mesmo aquelas entidades que descrevem outras entidades, são objetos do modelo.* Olhando o mundo desta forma, fica claro que os tipos de objetos (isto é, as classes) são também objetos do modelo, correspondendo aos conceitos abstratos do mundo real mencionado acima. Assim, não deveria haver nenhuma diferença entre os assim chamados tipos de objetos e as instâncias de objetos. Ambos são tratados como objetos do modelo. Assim, a diferença feita pelos sistemas de bancos de dados entre tipos de objetos incorporados ao esquema do banco de dados e as instâncias no banco de dados é eliminada aqui. Tanto os tipos quanto as instâncias serão tratadas como objetos do modelo, tal que as definições das estruturas estáticas como meta-informação não existirá. Semanticamente, pode haver objetos representando tipos ou objetos representando instâncias. Pode mesmo existir objetos que representem um tipo em um determinado contexto e que representem uma instância em outro contexto. Contudo, o modelo deve tratar todos os objetos da mesma forma mesmo que ele conheça exatamente o significado semântico de cada um deles em cada contexto particular. Para o engenheiro de conhecimento, é portanto possível aplicar as mesmas operações de manipulação de instâncias sobre tipos de objetos.

É também importante que as *entidades no mundo da aplicação correspondam a objetos no modelo de conhecimento*. Mais precisamente, uma correspondência um-a-um entre as entidades no mundo e os objetos no modelo deve existir, prevenindo assim, por exemplo, que o conhecimento sobre uma entidade (por exemplo uma pessoa) seja espalhado em vários objetos do modelo. Esta correspondência é o segundo axioma importante.

No mundo real, e conseqüentemente no modelo de conhecimento, *um objeto pode existir sem possuir uma "chave" (isto é, um identificador externo único) e ainda assim ser distinto de outros objetos*. Em outras palavras, é claro que um objeto não é a mesma coisa que o nome que o denota. Conseqüentemente, relacionamentos podem existir entre objetos e não entre seus nomes, identificadores ou descrições.

Cada objeto é composto de atributos que expressam as propriedades das entidades. Por exemplo, o objeto auto1 (veja a figura 3) possui os atributos cor, dono, lugar, e preço, denotando uma entidade do mundo real possuída por Maria, que tem cor branca, um preço de U$1000, e que neste momento está estacionado na Quinta Avenida.

Objeto auto1	
cor:	branco
dono:	Maria
lugar:	Quinta Avenida
preço:	U$ 1000,oo

Objeto auto2	
cor:	preto,branco
dono:	companhia de taxi
lugar:	Quarta Avenida
preço:	U$ 2500,oo
numeroDePassageiros:	3
motorista:	João

Objeto auto3	
cor:	branco,vermelho,verde
dono:	companhia de transporte
lugar:	desconhecido
preço:	U$ 5200,oo

FIGURA 3: Exemplo de uma descrição de propriedades de objetos.

Os atributos podem ser melhor descritos (refinados) depois. Isto permite expressar restrições sobre os atributos bem como incorporar mais semântica à base de conhecimento (figura 4). Descrições típicas a ser suportadas são o tipo de dados do valor do atributo (valores possíveis), número mínimo e máximo de valores permitido (cardinalidade) e valores por falta (default). Por exemplo, por definição, o proprietário de um carro deve ser uma pessoa ou uma instituição, um gerenciador de base de conhecimento "conhece" mais alguns detalhes sobre o mundo da aplicação do que apenas a interpretação do dono de um carro como uma cadeia de caracteres. Tal especificação permite ao modelo inferir que proprietários de carros são sempre pessoas ou instituições quando lhe for perguntado sobre isto. Note-se que a semântica da base de conhecimento é firmemente relacionada à forma de sua capacidade de raciocínio.

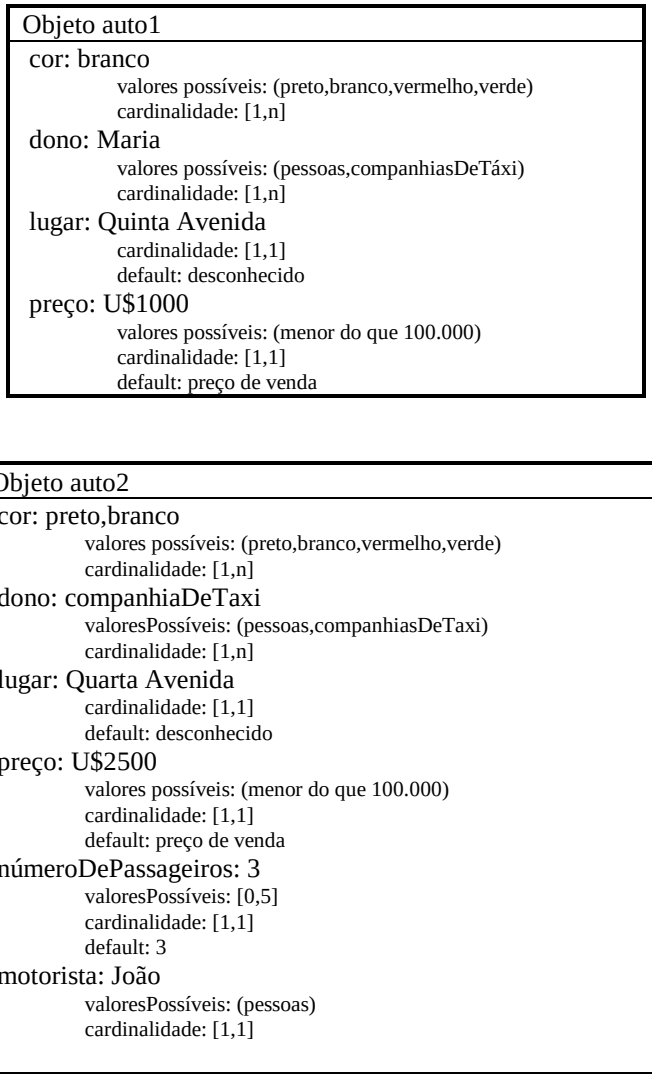


FIGURA 4: Exemplo de descrições de atributos.

Atributos de objetos podem ser multi-valorados (por exemplo, a propriedade cor na figura 3). Mais ainda, além dos tipos de dados tradicionais (inteiro, cadeia e booleano), outros como listas, tabelas e textos também poderiam ser suportados já que eles são típicos em domínios de aplicação de sistemas de conhecimento.

O último axioma diz que *relacionamentos entre entidades do real são suas propriedades, e podem, por esta razão ser expressos como atributos dos objetos*. Isto torna fácil a localização de objetos relacionados através do percorrimento das cadeias de propriedades sem ter que introduzir explicitamente objetos intermediários, como é feito em muitos modelos de dados. Por exemplo, a propriedade "dono" da entidade auto1 expressa o relacionamento entre a entidade e a pessoa chamada Maria (veja figura 3).

6.3 Aspectos Operacionais

Os aspectos operacionais estão envolvidos com a representação das características procedurais do domínio de aplicação bem como das estratégias de solução dos especialistas humanos.

6.3.1 Comportamentos

Na seção anterior foi visto o lado passivo do domínio da aplicação. Ele foi definido como uma coleção de objetos inativos que têm somente características declarativas associadas a eles. Entretanto, entidades do mundo real também têm características procedurais tal que o domínio de aplicação deve ser, na verdade, definido como uma coleção de objetos ativos, ou atores. A primeira característica procedural das entidades é o seu comportamento, isto é, o conhecimento sobre o comportamento das entidades em geral. Por exemplo, o conhecimento sobre as ações que tomam lugar quando um carro é movido define um aspecto particular de seu comportamento.

Quando se está modelando o domínio da aplicação, tais comportamentos, bem como outras características procedurais das entidades do mundo real são muito importantes, porque elas estão relacionadas intrinsecamente com as propriedades declarativas destas entidades. Comportamentos, por exemplo, geralmente realizam transformações nas propriedades declarativas de uma entidade quando ativados, definindo portanto o comportamento desta entidade em uma situação particular. Por exemplo, o comportamento de "deslocamento" dos carros, quando ativado, muda a quantidade de gasolina do tanque bem como a quantidade de quilômetros rodados pelo carro, exatamente como ocorre no mundo real.

Portanto, comportamentos são também propriedades das entidades e devem, por esta razão ser representados como atributos dos objetos e tratados exatamente da mesma forma que qualquer atributo descritivo. Conseqüentemente, a maioria dos axiomas estabelecidos na seção anterior sobre objetos e seus atributos são também válidos aqui. Desta maneira, uma visão integrada dos automóveis descritos anteriormente ainda apresenta a estrutura discutida exceto por alguns atributos novos que expressam as propriedades procedurais destes carros (figura 5).

Objeto auto2

cor: preto,branco
- valores possíveis: (preto,branco,vermelho,verde)
- cardinalidade: [1,n]

dono: companhiaDeTaxi
- valoresPossíveis: (pessoas,companhiasDeTaxi)
- cardinalidade: [1,n]

lugar: Quarta Avenida
- cardinalidade: [1,1]
- default: desconhecido

preço: U$2500
- valoresPossíveis: (menor do que 100.000)
- cardinalidade: [1,1]
- default: preço de venda

númeroDePassageiros: 3
- valoresPossíveis: [0,5]
- cardinalidade: [1,1]
- default: 3

motorista: João
- valoresPossíveis: (pessoas)
- cardinalidade: [1,1]

quantidadeDeGasolina: 32
- valoresPossíveis: [0,40]
- cardinalidade: [1,1]
- unidade: litros

KMrodados: 42575
- valoresPossíveis: maior do que 0
- cardinalidade: [1,1]
- **rodar (mudar lugar, aumentar KMrodados, diminuir quantidadeDeGasolina)**
- unidade: quilômetros

FIGURA 5: Exemplo de uma descrição de atributos comportamentais.

A ativação de comportamentos ocorre, como no mundo real, por interações entre objetos. Por exemplo, quando João quer que seu táxi ande, ele interage com ele ligando seu motor e dirigindo-o. O componente ativado pela interação, isto é, o receptor, especifica os tipos de ações que serão tomadas. Como uma propriedade de si mesmo, o receptor "conhece" as ações associadas com seu comportamento. Observe que João não precisa conhecer nada sobre as ações que ocorrem no processo de dirigir o seu táxi. A única coisa que ele precisa conhecer é como interagir com ele, com o objetivo de colocá-lo em movimento. Uma vez colocado em movimento, seu táxi trabalhará por si só, desde que João o dirija corretamente.

Normalmente, comportamentos não provocam somente mudanças em algumas propriedades dos objetos, mas também ativam comportamentos de outros objetos. Por exemplo, rodar com um carro, certamente ativará muitos outros comportamentos dos componentes (isto é, motor, rodas, etc) do carro.

6.3.2 Reações

O segundo tipo de conhecimento procedural importante encontrado no domínio da aplicação são as reações das entidades do mundo real quando situações particulares ocorrem: por exemplo, a reação de João quando um carro em frente a seu táxi subitamente freia impetuosamente. Quanto se descreve tal conhecimento, não se está interessado em caracterizar comportamentos dos objetos mas as conseqüências de eventos do mundo real.

É claro, entretanto, que tais conseqüências somente podem ter efeito nas entidades do mundo real. Apesar disso, elas são traduzidas tanto em ativações de comportamentos de objetos quanto em modificações de seu conhecimento declarativo. No exemplo acima, uma conseqüência possível do evento descrito poderia ser a ativação do comportamento "freiar" do carro de João tal que nada maisaconteça. Outra conseqüência poderia ser o próprio acidente que causaria uma série de mudanças na descrição de cada um dos objetos envolvidos (os dois carros, João, o outro motorista, etc).

Outro exemplo de tal tipo de conhecimento é a reação de João quando o tanque de seu táxi está quase vazio. Neste caso, a ação de encher o tanque do carro (isto é, um comportamento de João) será tomada.

É importante observar que a descrição de situações do mundo real constitui conhecimento declarativo. Por esta razão, as reações correspondentes são fortemente relacionadas aos aspectos descritivos do modelo de conhecimento e devem ser, conseqüentemente ligadas aos objetos e/ou propriedades dos objetos. Reações são, portanto, descrições adicionais dos objetos e seus atributos e devem ser tratadas simplesmente como outras descrições.

Assim, a reação sobre um tanque vazio é encontrada sob as descrições da propriedade "quantidadeDeGasolina" no táxi de João (figura 6).

Objeto auto2
cor: preto,branco valores possíveis: (preto,branco,vermelho,verde) cardinalidade: [1,n] dono: companhiaDeTaxi valoresPossíveis: (pessoas,companhiasDeTaxi) cardinalidade: [1,n] lugar: Quarta Avenida cardinalidade: [1,1] default: desconhecido preço: U$2500 valoresPossíveis: (menor do que 100.000) cardinalidade: [1,1] default: preço de venda númeroDePassageiros: 3 valoresPossíveis: [0,5] cardinalidade: [1,1] default: 3 motorista: João valoresPossíveis: (pessoas) cardinalidade: [1,1] quantidadeDeGasolina: 32 valoresPossíveis: [0,40] cardinalidade: [1,1] unidade: litros reação: com tanque vazio, ativar motorista para encher. KMrodados: 42575 valoresPossíveis: maior do que 0 cardinalidade: [1,1] rodar (mudar lugar, aumentar KMrodados, diminuir quantidadeDeGasolina)

FIGURA 6: Exemplo de uma descrição de uma reação.

6.3.3 Regras Situação-Ação

Até este ponto, as construções que foram analisadas estavam envolvidas com a descrição do domínio de conhecimento expressado por meio de objetos, atributos, ações, relacionamentos e restrições. O modelo de conhecimento, entretanto, também tem construções para representar o conhecimento de resolução de problemas dos especialistas humanos sobre este domínio de aplicação.

Empiricamente, parece que os especialistas expressam a maior parte de seu conhecimento sobre solução de problemas simplesmente como regras de situação-ação. Situações são usualmente associadas com a descrição de algum estado do mundo real, e a maioria das ações determinam algum tipo de conclusão do especialista sobre este estado. Assim, situações bem como ações são de natureza passiva. Elas definem aspectos descritivos do domínio da aplicação e são, por esta razão um tipo de conhecimento declarativo. Apesar disso, o aspecto mais importante de tais situações e ações não é o seu conteúdo declarativo, mas o efeito procedural resultante da combinação de ambas, isto é, "quando a situação X ocorrer, então a ação Y será tomada". Este aspecto procedural é fortemente relacionado ao processo de raciocínio e deve ser suportado pelo modelo de conhecimento. Isto significa que o modelo de conhecimento deve primeiramente fornecer construções para permitir a representação de tais regras, e em seguida os mecanismos para interpretá-las (isto é, para fazer deduções).

Outro aspecto importante de tais regras é que elas são um tipo de meta-conhecimento, já que elas definem informações sobre o domínio de conhecimento. Tanto o domínio quanto o conhecimento do especialista são assim conectados mas não são dependentes um do outro. Note-se,

por exemplo, que o domínio do conhecimento pode ser expresso sem a representação de qualquer conhecimento de especialista sobre ele. Por esta razão, eles devem ser modelados independentemente um do outro, permitindo modificações bem como expansão incremental de qualquer um sem afetar o outro. O modelo e conhecimento deve, entretanto, oferecer maneiras de referir os objetos do domínio, seus atributos, e descrições quando forem especificadas regras situação-ação.

6.4 Aspectos Organizacionais

As pessoas raramente descrevem o mundo real da maneira que foi discutida nas seções anteriores, isto e', em termos de informação "factual" específica sobre o domínio da aplicação. Não é nenhuma surpresa que eles usem construções que têm suas raízes em métodos epistemológicos para organizar conhecimento (isto é, o conhecimento estrutural sobre o domínio). Em outras palavras, quando se tenta modelar o mundo da aplicação, as pessoas geralmente aplicam involuntariamente algumas abstrações com o objetivo e organizar seu conhecimento em alguma forma desejada. A abstração permite que alguém suprima detalhes específicos em objetos articulares, enfatizando aqueles que pertencem ao problema ou visão da informação em mãos. Esta é, portanto, a ferramenta de modelagem fundamental para descrever conhecimento, e é, por esta razão a construção mais importante a ser suportada pelo modelo de conhecimento. Assim, é vantajoso focalizar em grande detalhe cada m dos conceitos de abstração existentes e, para formalizá-los, caracterizar seu significado, examinando sua aplicabilidade, e estudando sua utilidade para poder diferenciá-los um do outro.

Numerosos artigos tentaram descrever um ou outro conceito e vários modelos tem sido desenvolvidos seguindo alguma destas descrições. Entretanto, definições precisas da semântica e da aplicabilidade destes conceitos não existem. Adicionalmente, abordagens sobre uma integração de todos os conceitos em um único modelo são raramente encontradas. A maioria dos modelos desenvolvidos e concentram no suporte de um ou dois conceitos de abstração, ignorando a existência dos outros.

Nesta seção, nós nos concentramos em uma visão neutra da semântica e construções que existem por trás de tais conceitos de abstração. Em primeiro lugar, estes conceitos serão precisamente definidos pela caracterização de seu significado. Em seguida, será discutida a sua aplicabilidade pelo exame de facilidades de raciocínio construídas neles e será mostrada uma abordagem sobre integração destes conceitos de abstração. Enquanto lê esta seção, o leitor deverá ter em mente que os pontos discutidos previamente, isto é, os aspectos descritivos e operacionais do conhecimento ainda são válidos. Entretanto, para simplificar, os exemplos são baseados apenas nos aspectos descritivos dos objetos o mundo real.

Do ponto de vista dos conceitos de abstração, cada objeto do mundo real pode ser simples (isto é, definido por si só), ou composto (isto é, definido como uma abstração de outros objetos). abstrações são expressas como relacionamentos entre objetos, tendo como seu propósito a organização destes objetos de alguma forma.

Dois tipos de relacionamento de abstração podem ser encontrados no mundo real: um envolvendo objetos simples com o objetivo e construir um objeto composto, e outro envolvendo objetos compostos com o objetivo de construir outros objetos compostos mais complexos ainda. Abstrações do primeiro tipo são, portanto, relacionamentos de nível um, já que elas são aplicadas apenas uma vez ara construir os objetos compostos menos complexos. Abstrações o segundo tipo são, por sua vez, relacionamentos de nível n, já que elas podem ser aplicadas n vezes, permitindo

que os objetos compostos sejam representados, num certo sentido, recursivamente por outros menos complexos. A diferenciação destes dois tipos de relacionamento é bastante importante por causa da semântica especial que eles incorporam.

6.4.1 Classificação

Encontrada em quase todos os modelos de dados e sistemas de conhecimento existentes, a classificação é a mais importante e a ais bem entendida forma de abstração. Ela é obtida quando se agrupam objetos que tem propriedades em comum em um novo objeto ara o qual condições uniformes são cumpridas. Em outras palavras, classificação é uma forma de abstração na qual um objeto composto, chamado, neste contexto, de objeto *tipo* ou *classe*, é definido como um conjunto de objetos simples, os quais têm as mesmas propriedades, e são chamados *instâncias*. Isto estabelece m relacionamento *instância-de* entre alguns objetos simples (as instâncias) e um objeto composto (a classe). Classificação é, portanto um *relacionamento de abstração de nível um*.

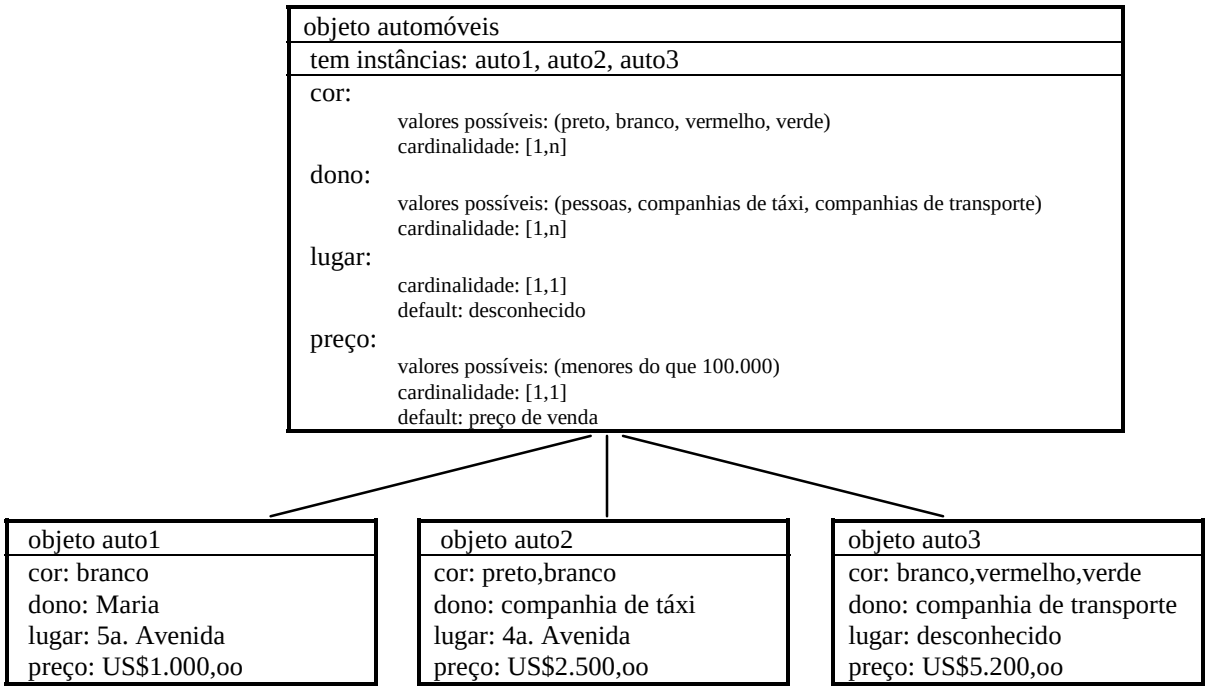
Por exemplo, se existem três objetos no mundo real, todos eles tendo as propriedades de cor, dono, lugar e preço (figura), alguém poderia descrever cada um deles independentemente, especificando alguns valores particulares de suas propriedades.

objeto auto1
cor: branco dono: Maria lugar: 5a. Avenida preço: US$1.000,oo

objeto auto2
cor: preto,branco dono: companhia de táxi lugar: 4a. Avenida preço: US$2.500,oo

objeto auto3
cor: branco,vermelho,verde dono: companhia de transporte lugar: desconhecido preço: US$5.200,oo

a) descrição factual.



b) descrição por classificação.

Figura 7.: Exemplo de classificação.

Entretanto, descrever um modelo em termos de informação "factual" específica dificilmente será satisfatório. Freqüentemente é importante abstrair os detalhes de cada objeto e tratá-los de uma maneira mais genérica, sem ter que se preocupar com os valores específicos de cada propriedade. De fato, a informação requerida de natureza genérica, isto é, um modo de referir a todos os objetos abstraindo os detalhes de cada um. A classificação provê um ecanismo importante para introduzir tal informação genérica por permitir que o modelador se refira à classe como um representante, ou protótipo, das instâncias. Na classe tanto as propriedades quanto as restrições aplicáveis a todas as instâncias estão representadas.

O inverso da classificação, a *instanciação*, pode ser usada, por outro lado, para obter objetos que satisfaçam as restrições associadas com as propriedades especificadas pela classe. Já que um objeto pode ser uma instância de mais do que uma classe, a instanciação pode ser aplicada para criar objetos que possuam as propriedades de ambas as classes (por exemplo, o objeto auto2 da figura 8).

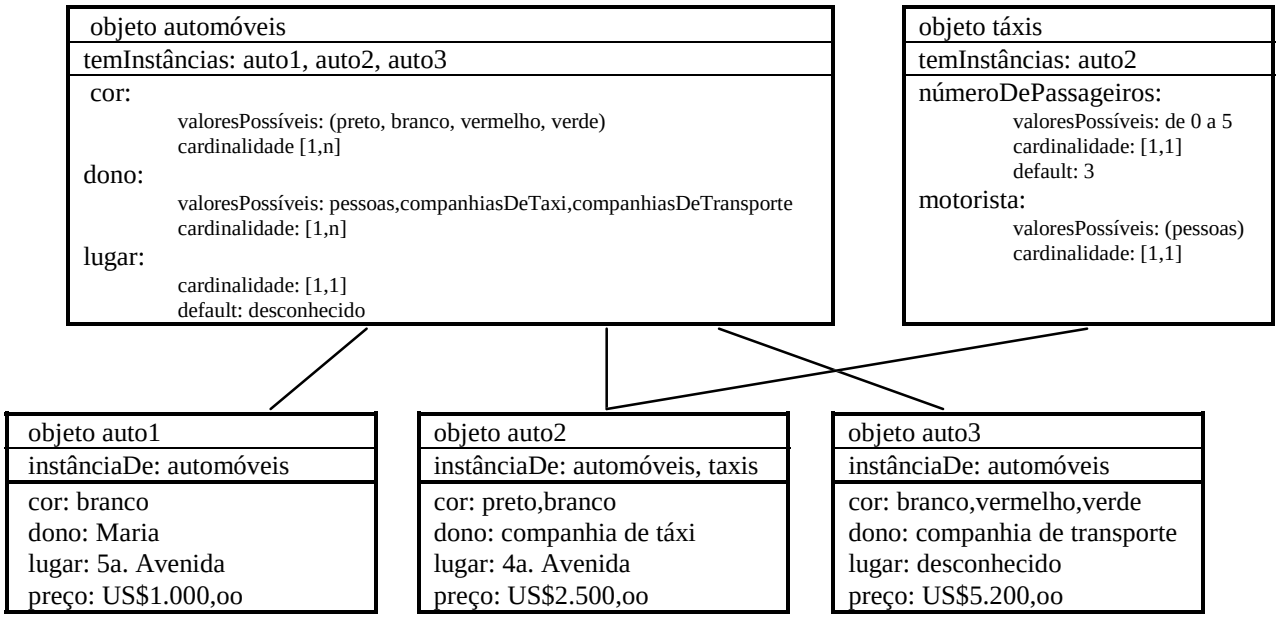
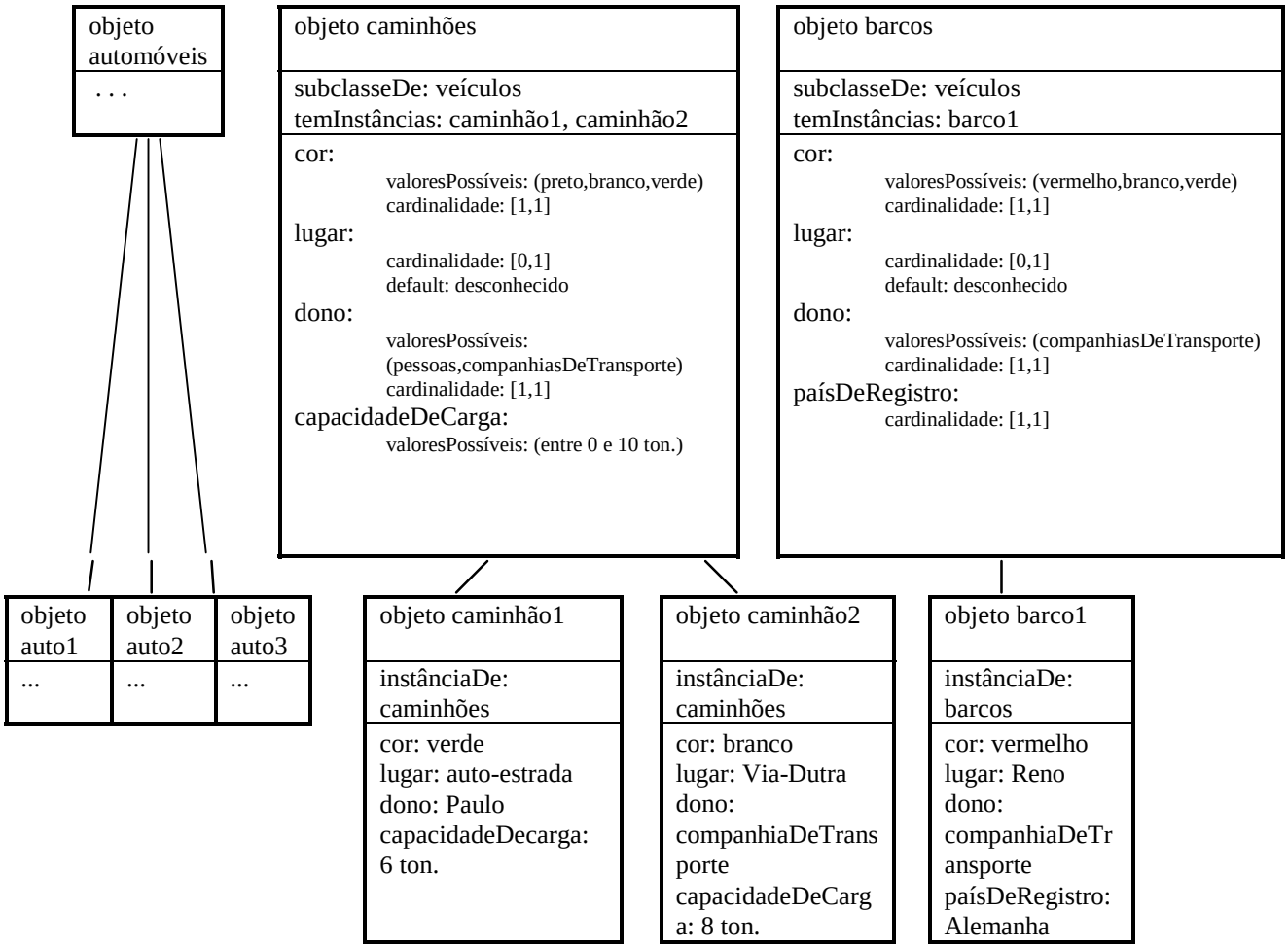


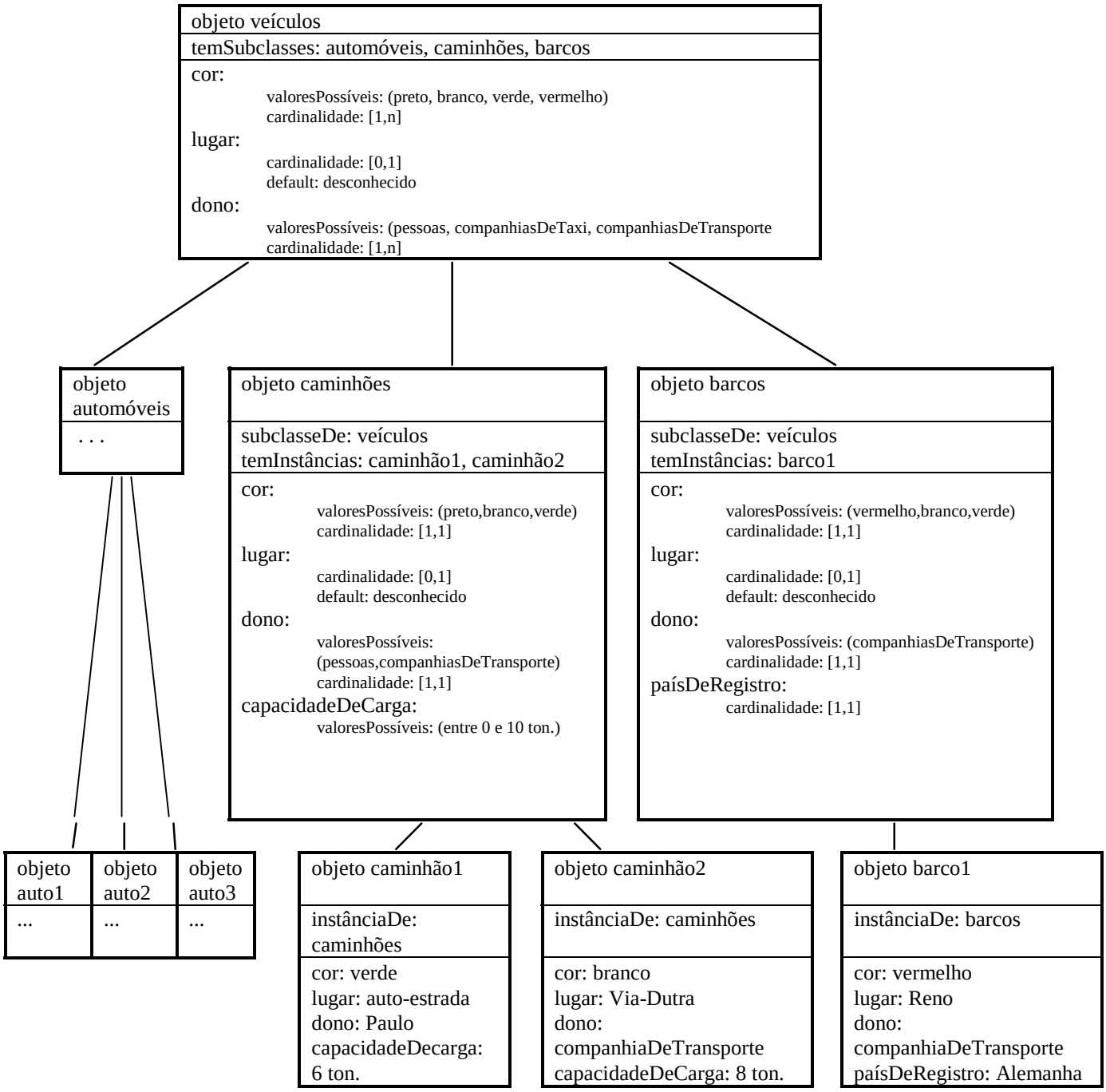
Figura 8: Exemplo de uma descrição de instâncias de múltiplas classes.

6.4.2 Generalização

Generalização é o conceito complementar à classificação. Digamos que em adição aos três automóveis descritos no último exemplo (figura 8), nosso mundo real também tenha caminhões (caminhão1, caminhão2) e barcos (barco1). A classificação poderia, então, prover a descrição mostrada da figura 9. Entretanto, da mesma forma que no caso da classificação, seria importante ter uma maneira de se referir a todas as três classes, ou mesmo, a todas as suas instâncias abstraindo os detalhes de cada uma. Em outras palavras, olha-se para um modo de extrair de uma ou mais classes dadas a descrição de uma classe mais geral que captura os aspectos comuns e suprime alguns dos detalhes diferenciados nas descrições das classes dadas. A generalização fornece este caminho, por permitir a definição de objetos compostos mais complexos, chamados *superclasses*, como uma coleção de outros menos complexos, chamados *subclasses*, estabelecendo uma relação *é-um*, ou *subclasse-de* entre a superclasse e as subclasses. Neste exemplo, a classe veículos é, então, definida como uma generalização de automóveis, caminhões e barcos (figura 9).



a) descrição por classificação



b) descrição por generalização e classificação.

Figura 9: Exemplo de Generalização.

Já que a generalização pode ser aplicada recursivamente para especificar outras superclasses de nível mais alto, ela é um *relacionamento de nível n* que organiza as classes em uma *generalização* ou *hierarquia é-um*. Por exemplo, veículos juntamente com aviões podem ser generalizados como meiosDeTransporte (figura 10). Como conseqüência, uma grande homogeneidade entre certas classes é obtida, já que classes que têm superclasses em comum podem diferir em apenas poucos detalhes. Isto normalmente não é típico para aplicações convencionais em bancos de dados, onde a regra é a existência de uma grande não-homogeneidade entre os tipos dos objetos (isto é, classes) e, por outro lado, de uma grande homogeneidade dentro de um tipo de objeto (isto é, entre as instâncias correspondentes).

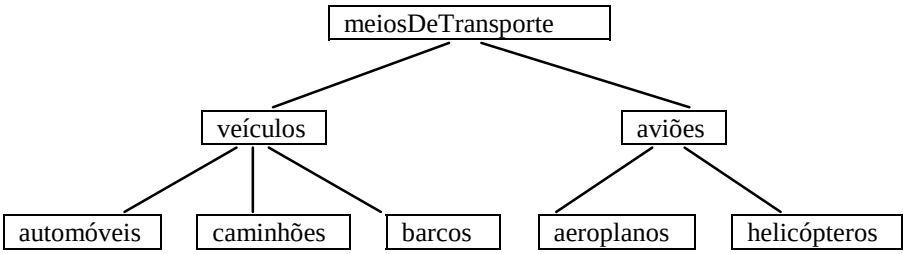


Figura 10: Exemplo de uma Hierarquia de Generalização.

A metodologia para modelar o mundo real, é, entretanto, baseada em um processo que tem o efeito exatamente oposto da generalização, isto é, a *especialização*. De acordo com esta metodologia, um modelo é construído primeiramente pela descrição das classes mais gerais e então procedendo com a descrição das subclasses até aquelas mais especializadas. Esta metodologia utiliza a mais importante vantagem do conceito de generalização, ou seja, a herança de propriedades.

6.4.3 Herança

Já que as propriedades estabelecidas por superclasses são geralmente válidas também para suas subclasses, não há necessidade de repetir a descrição armazenada na superclasse para cada uma de suas subclasses. Como resultado disto, as descrições podem ser abreviadas sem perder a necessária flexibilidade para descrever as propriedades dos objetos, e alguns erros clássicos podem ser evitados pela redução da quantidade de repetições nas descrições.

Há duas formas básicas de ver a descrição associada a uma classe, e conseqüentemente de tratar a herança. A primeira, que é a forma que os sistemas de representação de conhecimento normalmente vêem as classes, é que as classes simplesmente caracterizam instâncias prototípicas. Como conseqüência, as propriedades herdadas podem ser contraditas pelas subclasses de uma classe, ou mesmo por instâncias particulares da mesma. A segunda, que representa a abordagem seguida pelos modelos de dados semânticos, vê a descrição associada com uma classe como as condições necessárias (isto é, propriedades e restrições) que cada instância ou subclasse deve satisfazer. As conseqüências destas duas visões seguem imediatamente:

a) Se classes são tratadas em um sentido prototípico, a herança é feita por default (herança default), isto é, propriedades herdadas são válidas somente se elas não forem contraditas na definição de uma subclasse ou instância. Por exemplo, mesmo que exista uma propriedade na descrição da classe aves que diga que todas as aves voam, e pingüins são uma subclasse de aves, pingüins não voam necessariamente.

b) Se as classes são tratadas como abstrações (templates), a herança se torna estrita, isto é, todas as propriedades herdadas devem existir e serem válidas para cada subclasse, e conseqüentemente para cada instância. Neste caso, não seria possível definir uma propriedade que dissesse que todas as aves voam (na classe das aves), já que esta propriedade não é satisfeita por todas as subclasses de aves.

Em uma primeira olhada, o sentido prototípico de classes parece ser o mais apropriado, uma vez que ele permite uma representação direta de exceções. Entretanto, ele enfraquece a força assercional associada às definições de classes. Quando se permite que propriedades herdadas sejam canceladas em uma subclasse, a classificação não pode ocorrer entre as instâncias das subclasses e

suas superclasses. Por exemplo, não é possível afirmar que as instâncias de pingüim (por exemplo, Picolino) sejam também instâncias de aves, já que eles não podem satisfazer todas as restrições de propriedades especificadas para aves (veja a definição de classificação). Analisando a semântica do sentido de templates mais a fundo, pode-se perceber que ele não apenas suporta maneiras para representar exceções, mas também fortifica a semântica do conceito de classificação. Aqui, exceções podem ser expressas pelo refinamento da definição de uma propriedade quando uma classe é especializada. Por exemplo, na superclasse Aves a propriedade de locomoção poderia expressar que aves voam, nadam ou ambos, e na subclasse dos pingüins, esta propriedade poderia ser refinada tal que a locomoção dos pingüins fosse restrita a "nadar". Conseqüentemente, todas as restrições associadas às superclasses são válidas para as subclasses já que elas ou são a mesma, ou são mais restritivas (por exemplo, a propriedade da cor na figura 9). Naturalmente, todas instâncias das subclasses são também instâncias das superclasses, fortificando assim o conceito de classificação.

Por todas estas razões, nós adotamos a segunda visão da classificação (b). Assim, pode-se afirmar que as únicas mudanças possíveis às propriedades herdadas são aquelas que restringem mais ainda as restrições associadas a elas.

6.4.4 Resumo

Sumarizando a idéia da classificação, pode-se dizer que um objeto se torna uma instância de uma classe pela adição de valores que satisfazem as restrições das propriedades da classe. Em outras palavras, o relacionamento instânciaDe é restrito a mudanças nos valores das propriedades, requerindo que tudo o mais seja deixado constante.

Generalização é um relacionamento de n-níveis entre objetos compostos (superclasses e subclasses), que (juntamente com a classificação) constrói uma hierarquia é-um. As subclasses têm exatamente as mesmas propriedades das superclasses, neste caso, com restrições mais específicas (por exemplo, cor), ou novas propriedades em adição às herdadas (por exemplo, preço). Em outras palavras, um relacionamento é-um é restrito à adição de novas propriedades ou a especialização de restrições, requerindo que tudo o mais seja deixado constante. Por esta razão, a classificação também ocorre entre superclasses e instâncias de subclasses tal que a generalização pode ser vista como complementar à classificação.

6.4.5 Associação-Elemento

Freqüentemente é necessário agrupar objetos por outros motivos que não sejam a co-posse de propriedades comuns (classificação), mas porque é importante descrever propriedades de um grupo de objetos como um todo.

Digamos que em nosso exemplo de transporte que cada veículo (isto é, automóveis, barcos e caminhões) tenha uma propriedade adicional para expressar seus custos. Vamos também assumir que alguém está interessado em representar o conjunto de veículos de alguma companhia de transporte para expressar propriedades de sua frota como, por exemplo, seu custo médio. Já que tal propriedade só pode existir em associação com objetos (veja o primeiro e o quarto axiomas da seção 1), é necessário definir um novo objeto que possui esta propriedade.

A primeira idéia que se poderia aplicar seria definir uma nova classe que tem os veículos de uma companhia como instâncias. Entretanto, se a definição da classificação for analisada mais uma vez, se perceberá que este modelo não é consistente. Uma classe contém a definição das propriedades e restrições associadas que cada uma de suas instâncias deve respectivamente conter e satisfazer. Conseqüentemente, cada instância de veículosDeCompanhiaDeTransporte (isto é, auto3,

barco1 e caminhão2) teriam que ter a propriedade custoMédio. Esta é, entretanto, uma propriedade que estes objetos não possuem e não precisam satisfazer por sua vez. Portanto, não é uma propriedade de cada objeto, mas do grupo como um todo.

Para eliminar este problema, tem-se que usar o conceito de associação-elemento. De acordo com ele, um outro tipo de objeto, chamado *conjunto* é introduzido para representar este tipo de propriedades. Em outras palavras, associação-elemento é uma forma de abstração na qual alguns objetos, chamados *elementos*, são considerados como um objeto de nível mais alto. Os detalhes dos objetos elementos são suprimidos e as propriedades do grupo são enfatizadas no objeto conjunto, estabelecendo um *relacionamento elementoDe* entre elementos e conjunto.

6.4.6 Associação-Conjunto

Como se pode ver, o conceito de associação-elemento, também chamado agrupamento, particionamento ou agregação por cobertura, corresponde a uma parte da teoria matemática dos conjuntos. Portanto, em adição ao relacionamento nível-um descrito acima, o qual constrói objetos compostos (conjuntos) baseados em objetos simples (elementos), é também necessário ter uma maneira de expressar relacionamentos entre objetos compostos (conjuntos) com o objetivo de suportar a teoria dos conjuntos completa. Estes relacionamentos estão embutidos no assim chamado conceito conjunto-associação, onde eles são denotados por r-Ielacionamentos subconjuntoDe.-z Isto corresponderia a dizer que o conjunto veículosDeCompanhiaDeTransporte é um subconjunto do conjunto meiosDeTransporteDeCompanhiasDeTransporte, o qual reúne não apenas os veículos da companhia, mas também seus aviões. Desta maneira, o relacionamento associação-elemento corresponde à noção matemática da relação de pertença, enquanto que a associação-conjunto corresponde a relação de inclusão. Assim, o relacionamento subconjunto-de é nível-n, já que ele pode ser aplicado recursivamente, definindo (juntamente com o relacionamento elemento-de) uma hierarquia de objetos, também chamada de *hierarquia de associação* (figura 11).

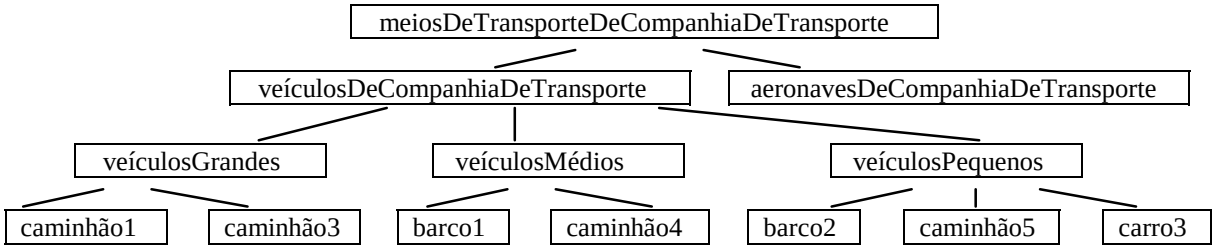


Figura 11: Exemplo de uma hierarquia de associação.

Similarmente à asserção feita na hierarquia é-um, de que toda instância de uma classe é também instância das superclasses da classe, elementos de um conjunto são também elementos dos superconjuntos do conjunto. Da mesma forma, objetos podem ser elementos de vários conjuntos e conjuntos podem ser subconjuntos de vários outros.

6.4.7 Propriedades de Conjuntos e Condições de Pertinência

Certamente, não há herança na associação, já que as propriedades descritas nos conjuntos não são características dos elementos. De fato, elas expressam propriedades do conjunto como um todo, as quais são determinadas baseadas em propriedades dos elementos, por exemplo: média, e valores máximo e mínimo.

Em adição a este conjunto de propriedades, descrições de conjuntos contém outras propriedades que são válidas para cada elemento do conjunto. Por exemplo, todo elemento do

conjunto veículosDeCompanhiaDeTransporte têm a companhia como valor da propriedade "dono". Nós denotamos estas propriedades como condições de pertinência.

Um objeto conjunto pode se tornar um subconjunto de outro pela restrição de alguma condição de pertinência. Por exemplo, elementos do conjunto veículosDeCompanhiaDeTransporte tem como valor da propriedade capacidadeDeCarga alta, média ou baixa, enquanto que o valor dos elementos do subconjunto veículosGrandes é restrito a alto (figura 11). Quando se restringe condições de pertinência, o cálculo das propriedades do conjunto será conseqüentemente restrito a um subgrupo dos elementos do superconjunto, determinando valores diferentes para estas propriedades. Por exemplo, o cálculo da propriedade custosMédios do conjunto veículosGrandes será restrito apenas aos caminhões grandes, e não à frota inteira.

6.4.8 Resumo

Associações relacionam objetos através de relacionamentos elementoDe e subconjuntoDe, construindo uma hierarquia de associação. Um objeto se torna um subconjunto de outro quando suas condições de pertinência são restritas, as quais, por sua vez, restringem os elementos usados para calcular as propriedades do conjunto. Objetos são elementos de conjuntos quando eles satisfazem as condições de pertinência dos conjuntos.

6.4.9 Agregação-Elemento e Agregação-Componente

Freqüentemente, é necessário tratar objetos não como entidades atômicas, como se fazia em classificação, generalização e associação, mas como uma composição de outros objetos. Vamos examinar, por exemplo, o objeto automóveis. Seria importante, para uma aplicação particular expressar o fato de que um automóvel consiste de um motor, rodas e uma carroceria, suprimindo, entretanto, em um primeiro momento, os detalhes específicos dos objetos constituintes (isto é, que estes objetos possuem também propriedades e também consistem de outros objetos). Isto está de acordo com o conceito de abstração por agregação, de acordo a qual uma coleção de objetos é vista como um único objeto de maior nível. Automóveis são tratados, portanto, como uma agregação dos objetos motor, rodas e carroceria.

Assim como os outros conceitos, a agregação envolve dois tipos de objetos: simples e compostos. Os objetos simples, chamados *elementos*, são atômicos na agregação, isto é, eles são objetos que não podem mais ser decompostos. Juntos, eles correspondem aos objetos compostos de mais baixo nível, ou simplesmente componentes. Estes, por sua vez, podem ser usados para construir componentes mais complexos de mais alta ordem tal que uma *agregação*, ou *hierarquia parte-de* é formada (figura 12). Entre objetos componentes há um *relacionamento subcomponente-de*, ao passo que entre componentes e elementos existe um relacionamento *elemento-parte-de*. Relacionamentos subcomponente-de estabelecem o conceito elementar que nós denotamos agregação-componente de acordo com associação-conjunto e generalização, ao passo que relacionamentos elemento-parte-de estabelecem o conceito agregação-elemento de acordo com associação-elemento e classificação. Naturalmente, o relacionamento subcomponente-de é uma continuação de elemento-parte-de. Assim, elementos de subcomponentes são também elementos dos supercomponentes dos subcomponentes.

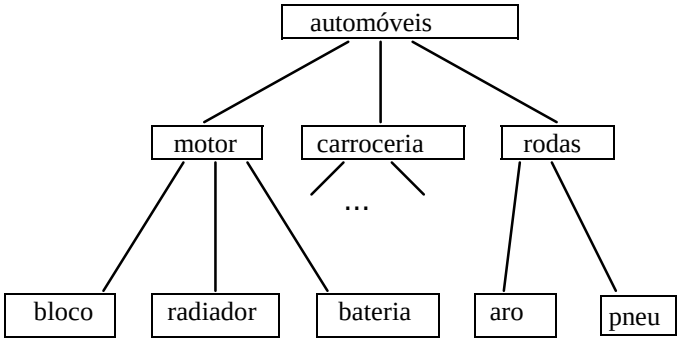


Figura 12: Exemplo de uma hierarquia de agregação.

6.4.10 Propriedades de Agregação

O conceito de agregação corresponde à noção de propriedade no sentido de composição uma vez que ele expressa a idéia de que automóveis "consistem-de" um motor, rodas e carroceria. Mais rigorosamente, podemos dizer que o conceito de agregação descreve propriedades necessárias que um objeto deve possuir para existir consistentemente. Em outras palavras, objetos compostos não podem existir sem seus elementos parciais. Com certeza, é difícil imaginar um automóvel sem motor. Note que esta necessidade torna o conceito de agregação bem diferente dos outros conceitos. Asclasses podem existir sem suas instâncias, e conjuntos vazios podem existir em uma associação. Propriedades de objetos agregação, entretanto, ao contrário dos outros conceitos, não são interpretadas como características dos mesmo, mas de outros objetos que representam suas partes. Normalmente estas propriedades não mudam com o tempo, já que objetos dificilmente mudam suas partes. Note que automóveis são sempre construídos com os mesmos componentes. Além disso, objetos com componentes diferentes provavelmente são objetos diferentes. Se alguém disser, por exemplo, que seu objeto não é uma agregação de motor, rodas e carroceria, mas uma agregação de motor, asas e cabine, poderia-se dizer que não se está descrevendo um automóvel, mas um avião.

6.4.11 Resumo

O conceito de agregação trata uma coleção de elementos como propriedades de um único objeto (componente) de mais alto nível. Estas propriedades, também chamadas propriedades parte-de, expressam relacionamentos que normalmente não mudam com o tempo, e devem ter um valor não-nulo com o objetivo de caracterizar os objetos completamente. Aplicando a agregação recursivamente sobre os objetos constrói-se uma hierarquia de agregação, tendo os objetos atômicos (elementos) como suas folhas.

6.4.12 Facilidades de Raciocínio Embutidas

Como foi visto na seção 3.1, os conceitos de abstração fornecem diferentes meios para organizar objetos: as hierarquias é-um, de associação e de agregação. Embora esta característica organizacional seja um aspecto significativo dos conceitos de abstração, a vantagem mais importante dos relacionamentos de abstração são as facilidades de raciocínio embutidas que eles fornecem. Quando se organiza objetos em hierarquias de abstração, diferentes facilidades de raciocínio "automático" são suportadas como parte de cada operação de modificação e recuperação.

6.4.13 Raciocínio Classificação/Generalização

A mais importante e útil destas facilidades de raciocínio é a que se constrói na hierarquia é-um, isto é, a herança. Ela permite que informações descritivas sejam compartilhas entre múltiplos objetos, determinando a estrutura interna das instâncias e provendo os significados para a manutenção automática de restrições de integridade semântica.

Esta facilidade de raciocínio é baseada na estrutura das classes (veja a seção 3.1.2). Desta maneira, quando se insere objetos no modelo expressando a crença do modelador de que eles são instâncias de alguma classe, o sistema consegue determinar exatamente a estrutura interna destes objetos. Por exemplo, quando se afirma que um objeto particular é uma instância de automóveis, o modelo de conhecimento pode recuperar a "crença" de que este objeto tem uma cor, um dono, um lugar e um preço. Da mesma forma, se a classe meiosDeTransporte possui a subclasse veículos, por exemplo, e veículos tem a subclasse automóveis, o modelo de conhecimento pode inferir que automóveis é uma subclasse de meiosDeTransporte, ainda que esta informação não esteja explicitamente representada no modelo de conhecimento. Conseqüentemente, o sistema inferirá também que instâncias de automóveis têm as propriedades especificadas em meiosDeTransporte e satisfazem as restrições associadas.

Seria importante mostrar em que ponto esta visão de classificação e instanciação difere daquelas suportadas pelos modelos de bancos de dados tradicionais. Em SBD's, a crença de que um objeto é uma instância de uma classe (por exemplo, uma tupla de uma relação) não está explicitamente representada no modelo, e não pode, por esta razão, ser verificada. De fato, a inserção de um objeto é condicionada à especificação do tipo do objeto (isto é, à especificação de exatamente um relacionamento instância-de). Em outras palavras, um objeto só pode existir se ele tiver um objeto-tipo (isto é, uma classe) implicitamente associado com ele. Adicionalmente, não é possível mudar o tipo de um objeto. Por exemplo, não é possível expressar no modelo relacional a crença de que um objeto auto1 não é uma instância de automóveis, mas de caminhões. A única maneira de fazer isto é deletando o objeto auto1 da relação automóveis e inserindo-o em uma nova relação, a dos caminhões, o qual, entretanto não será interpretada como o mesmo objeto auto1. Assim, o relacionamento instância-de é estático em SBD's convencionais (veja a seção 4.2.1.1).

De acordo com as observações acima, antes de inserir uma instância de objeto no modelo, um usuário de BD convencional deve primeiramente refletir sobre a estrutura deste objeto. O tipo (isto é, a relação) ao qual ele pertence deve ser também determinado pela determinação de sua aparência e a correspondente operação de inserção deve ser feita. Como uma conseqüência deste processo, que termina com a inserção de um objeto, um relacionamento instância-de entre o objeto inserido e seu tipo especificado será implicitamente criado. Por causa disso, descrições de classe (isto é, propriedades com restrições associadas) não podem ser usadas no mundo da base de conhecimento para deduzir dinamicamente "crenças" sobre as estruturas das instâncias, mas como condições auxiliares (no sentido da descrição de tipo) para evitar erros na inserção (ou na modificação) de objetos.

No modelo de conhecimento, os objetos podem existir por si só. É importante, portanto, permitir ao modelador a inserção de um objeto particular, mesmo quando ele não conhece o tipo do objeto (por exemplo, ele pode não estar certo, no momento da inserção, se auto1 é um carro, um caminhão ou outra coisa). Conseqüentemente, objetos podem existir no modelo independentemente da existência de relacionamentos instância-de. Freqüentemente, um objeto é primeiramente inserido no modelo sem a existência de qualquer especificação instância-de. Depois disso, o modelador pode refletir sobre o tipo deste objeto, ou ele pode querer ver como o objeto se comporta se ele for uma instância de uma classe particular (por exemplo, ele "acredita" primeiramente que auto1 é um automóvel) e define um relacionamento instância-de. (Note que no segundo caso o modelador estaria usando o SGBD para determinar a estrutura dos objetos, definindo dinamicamente os relacionamentos instância-de). O modelo de conhecimento deduzirá então a estrutura do objeto, gerando uma descrição mais detalhada dele. Baseado nesta nova descrição, o modelador pode agora perceber que ele não corresponde ao mundo real, iniciando a determinação do tipo do objeto mais

uma vez. Assim, relacionamentos instância-de correspondem, na verdade, a crenças reais do modelador, as quais podem mudar a qualquer momento sem afetar a existência dos objetos. Por esta razão, estes relacionamentos devem ser explicitamente representados no modelo de maneira que eles possam ser verificados a qualquer momento. Assim, especificações instância-de são aqui, em contraste com o mundo dos BD's convencionais, a causa e não a conseqüência no processo completo. Por causa disso, descrições de classe não podem ser usadas para evitar erros na inserção de objetos, como é feito nos BD's convencionais. Elas devem ser usadas para deduzir a estrutura destes objetos em conformidade com as crenças reais do modelador.

Outra facilidade de raciocínio pode usar a (já construída) estrutura de objetos para determinar a posição de um objeto dentro da hierarquia é-um. Baseado nas características dos relacionamentos subclasse-de e instância-de, o SGBC pode raciocinar analisando a estrutura de um dado objeto verificando quais objetos podem ser superclasses, subclasses e instâncias dele. Adicionalmente, quando usa as especificações de restrições, o modelo de conhecimento pode determinar se um dado item pode ser um valor de uma determinada propriedade, tal que os relacionamentos de generalização e/ou classificação são sejam violados. Por exemplo, quando um valor está sendo adicionado a uma propriedade de uma instância de objeto que já contenha um número máximo de valores permitidos, o valor deve ser ou rejeitado (porque caso contrário o relacionamento instância-de entre a instância e a classe na
qual a restrição é especificada não valerá mais), ou o relacionamento deve ser dissolvido. A decisão sobre que ação tomar deve ser feita pelo modelador. A única coisa que o SGBD pode fazer é relatar o estado inconsistente que seria provocado se ele aceitasse a modificação. (Note a diferença que há aqui, também, em relação aos SBD's convencionais.)

6.4.14 Raciocínio Associação

Como no caso de generalização/classificação, o poder do conceito de associação repousa nas "crenças" que o modelo pode inferir quando dois objetos estão relacionados pelo correspondente relacionamento de abstração. Por exemplo, quando se expressa que caminhão2 é uma instância-de caminhões (figura 9), o sistema pode inferir que ele também é um veículo, tem uma cor, um dono, um lugar e uma capacidadeDeCarga. Associações funcionam da mesma maneira. Por exemplo, se existe um conjunto veículosRápidos em nosso modelo, com a condição de pertinência de que todo elemento deste conjunto é um veículo e se move rápido, ao expressar que caminhão2 é um elemento deste conjunto, o sistema pode inferir que caminhão2 é um veículo rápido.

A diferença entre generalização/classificação e associação é que as propriedades de classes (isto é, aquelas que são especificadas pelas classes definindo a estrutura das instâncias) formam a estrutura dos objetos, enquanto condições de pertinência não formam. De fato, quando se restringe as condições de pertinência a que sejam expressas apenas pela estrutura dos objetos elemento, o poder do conceito de associação pode ser quase que totalmente perdido.

Vamos examinar primeiramente o significado das condições de pertinência. A conseqüência de expressar que caminhão2 é um elemento de veículos rápidos, isto é, que ele se move rápido, poderia também ser obtida definindo uma propriedade diretamente em caminhão2, a qual estabeleceria esta característica. Se veículosRápidos tem vários elementos, o mesmo pode ser feito através da definição de uma nova classe, na qual a propriedade velocidade rápida seria especificada, e poderia ser assim herdada por cada instância da classe. Assim, condições de pertinência correspondem a propriedades comuns com valores comuns que objetos heterogêneos (isto é, objetos que não tem superclasses comuns) possuem. A associação tem, portanto, em algum sentido uma herança embutida em seu conceito. Entretanto, o poder do conceito de associação repousa exatamente na não-hereditariedade destas propriedades.

Vamos assumir que seja necessário modelar as crenças de outro usuário que queira agrupar os veículosLentos. Vamos assumir que este segundo usuário é de opinião que caminhão2 é um veículo lento e afirmar que caminhão2 é um elemento de veículosLentos. Embora, a uma primeira vista, nosso modelo pareça ser inconsistente, já que caminhão2 é, ao mesmo tempo elemento de veículosRápidos e veículos lentos, isto não é inconsistente, já que cada um destes conjuntos representa crenças de duas pessoas diferentes. O modelo pode, entretanto, estar em um estado inconsistente se estas duas propriedades forem descritas pelo conceito de generalização/classificação e conseqüentemente herdadas por caminhão2, já que elas poderiam definir duas descrições estruturais para caminhão2. Note que, neste caso, caminhão2 teria diferentes descrições: uma com o valor rápido e outra com o valor lento para a mesma propriedade. (No mundo da inteligência artificial, este problema é conhecido pelo nome de conflito em herança múltipla.)

Entretanto, o significado da associação é outro. Quando não se aplica herança, não se definem descrições estruturais diferentes para caminhão2, mas diferentes "visões" dele. Isto é, caminhão2 é um veículo rápido (que é o mesmo que dizer que caminhão2 tem uma propriedade adicional que expressa que ele é rápido) visto do conjunto veículosRápidos. Entretanto, visto do conjunto de veículos lentos, caminhão2 é um veículo lento. (Note que condições de pertinência são propriedades que todo elemento deve satisfazer para ser um elemento do conjunto (veja a seção 3.1.2).

Associação, ao contrário de generalização/classificação, permite que o sistema deduza crenças que não estão restritas nem limitadas à estrutura dos objetos. Como conseqüência, muitos "mundos" diferentes, dos quais um pode contradizer o outro, podem ser facilmente modelados. Por causa disso, a associação é bastante útil na modelagem de atividades de planejamento e projeto, já que ela é muito apropriada para a representação de mundos hipotéticos e espaços de crenças.

Naturalmente, quando se especifica condições de pertinência baseadas na estrutura dos objetos elementos (por exemplo, elementos do conjunto veículosDeCompanhiaDeTransporte devem ter a companhia como valor da propriedade dono), o sistema deve determinar se uma mudança em um objeto elemento pode ser permitida, com o objetivo de não violar o relacionamento elemento-de e/ou subconjunto-de. É claro que o sistema deve rejeitar a mudança ou aceitá-la. Neste caso, o relacionamento elemento-de e/ou subconjunto-de violado pode ser dissolvido, e outros relacionamentos novos, os quais satisfazem o "novo" objeto podem ser criados. Contudo, o raciocínio baseado na estrutura dos objetos não é a meta do conceito de associação, mas da generalização/classificação.

Em adição a esta facilidade de raciocínio, a associação fornece outra facilidade que funciona em sentido diferente da herança. Isto é, enquanto as facilidades de raciocínio generalização/classificação dependem da estrutura das classes (isto e', elas fluem para baixo como a herança), a facilidade de raciocínio fornecida pelo conceito de associação é independente da estrutura dos conjuntos. Ao invés disso, as propriedades dos conjuntos são determinadas pelas propriedades dos elementos (isto é, elas fluem para cima). Assim, enquanto mudanças nas classes (isto é, a deleção de uma propriedade) implicam mudanças nos níveis inferiores do modelo, mudanças nas propriedades de um conjunto não causam nenhuma modificação nos objetos que são cobertos pelo conjunto. Entretanto, quando se modifica os elementos, mudanças nas propriedades superiores do conjunto são causadas. Por exemplo, quando se muda o valor da propriedade custo de um veículo de companhiaDeTransporte, o SGBC deve deduzir o novo valor correspondente da

propriedade custoMédio dos conjuntos veículosDeCompanhiasDeTransporte e meiosDeTransporteDeCompanhiasDeTransporte. Da mesma maneira, a dedução é necessária quando novos objetos se tornam elementos de um conjunto, para que se mantenha a consistência com a visão do mundo modelado.

6.4.15 Raciocínio Agregação

Da mesma forma que na associação, os relacionamentos de agregação não são dependentes da estrutura dos objetos. Eles suportam a noção de objetos complexos, o que permite ao usuário, em um determinado momento, ver um objeto como uma entidade única (isto é, juntamente com todas as suas partes) e em um outro momento ver apenas as partes como objetos independentes. Pela agregação, os objetos são vistos em um contexto como objetos mesmo e em outro contexto como componentes de outros objetos.

Entretanto, ao contrário da associação, as facilidades de raciocínio providas pelo conceito de agregação são baseadas nas estruturas dos objetos e, além disso, nas propriedades comuns dos componentes, sub-componentes e elementos. Estas facilidades de raciocínio são suportadas por predicados que são especificados sobre a hierarquia de agregação baseados nas propriedades dos componentes. Uma propriedade cujo valor diminui quando se desce na hierarquia de agregação, por exemplo, o peso (o peso de um subcomponente é sempre o mesmo, ou menor do que o peso de seus supercomponentes) é uma propriedade monotonicamente decrescente. Propriedades cujo valor aumenta quando se desce na hierarquia de agregação são monotonicamente crescentes.

Baseados nestas propriedades, os predicados são definidos de maneira que o raciocínio possa ser feito. Um predicado é implicado superiormente se o fato de que ele é satisfeito para um objeto implica que ele também é satisfeito por todos os seus supercomponentes. Por exemplo, o predicado "peso > 1 ton." é implicado superiormente já que todos os supercomponentes tem um peso maior do que qualquer um de seus sub-componentes. Um predicado é implicado inferiormente quando sua satisfação por um componente implica sua satisfação por todos os sub-componentes. Por exemplo, "tempoParaSerCompletado < Junho" é implicado inferiormente, já que todos os sub-componentes e elementos tem um tempoParaSerCompletado menor ou igual a qualquer de seus supercomponentes.

As facilidades de raciocínio do conceito de abstração correspondem às conclusões que o modelo de conhecimento pode fazer baseado nos predicados implicados quando dois objetos são relacionados pelo relacionamento subcomponente-de ou elemento-de. Por exemplo, conclusões sobre o peso de um componente podem ser feitas baseadas no correspondente predicado implicado superiormente. Da mesma maneira, o modelo de conhecimento pode usar estes predicados implicados como especificações de restrições para determinar se um valor dado pode ser o valor de alguma propriedade do objeto, de maneira que a monotonia desta propriedade não seja violada.

A última facilidade de raciocínio da agregação está relacionada ao axioma de que objetos não podem existir sem seus sub-componentes e elementos. Por causa disso, quando se deleta objetos, o modelo de conhecimento pode deduzir que todas as sub-partes deste objeto também foram deletadas. Por exemplo, quando um automóvel particular é removido de uma base de conhecimento, o sistema de conhecimento deduz que o motor, a carroceria e as rodas foram removidos juntamente com ele.

6.4.16 Integração dos Conceitos de Abstração

Até agora, nós isolamos a discussão sobre cada conceito de abstração. Nós vimos, por exemplo, que a associação constrói uma hierarquia de objetos a qual, entretanto, até este ponto, foi

construída independentemente da hierarquia é-um. O mesmo foi feito com a agregação. Assim, as três hierarquias existentes podem ser resumidamente esquematizadas na figura 13.

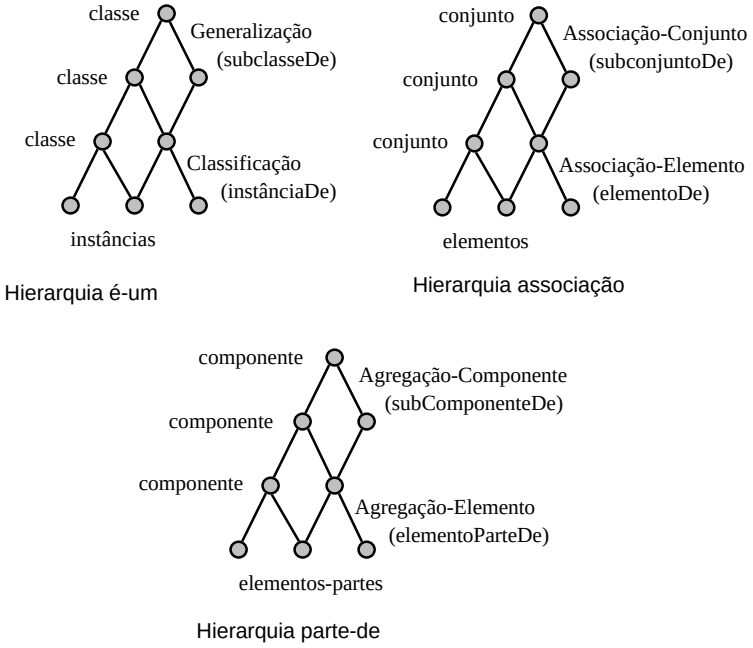


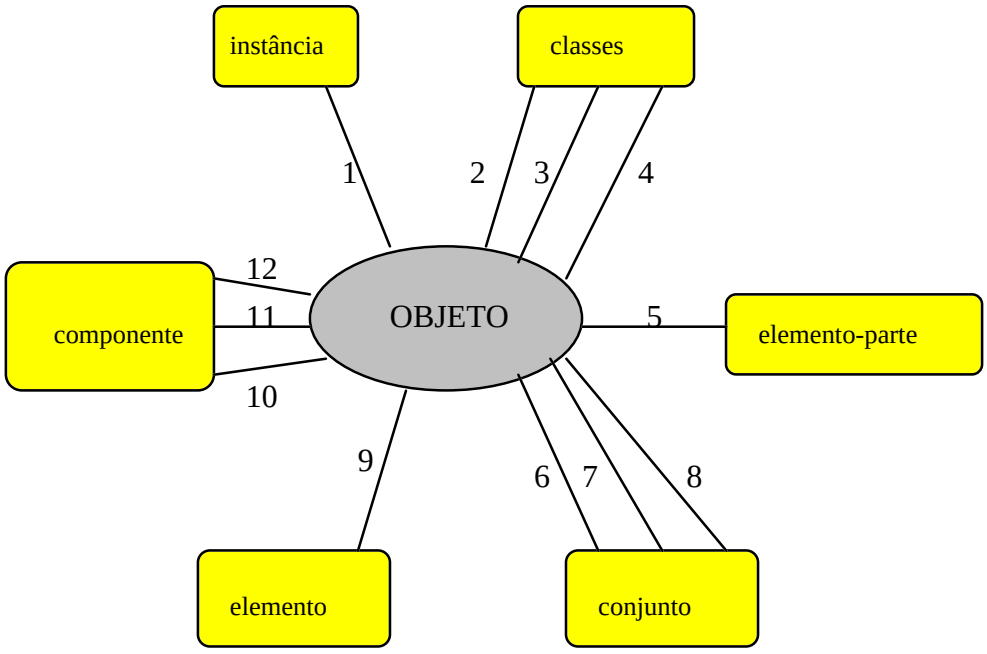
Figura 13: Esquema resumido dos conceitos de abstração.

Naturalmente, no mundo real, esta independência não existe. Objetos são instâncias de classes, elementos de conjuntos e componentes de agregações ao mesmo tempo. Por exemplo, os elementos do conjunto veículosDeCompanhiaDeTransporte são respectivamente instâncias de automóveis, barcos e caminhões (veja as figuras 9 e 11). O mesmo ocorre com o componente de uma agregação, por exemplo, um motor particular, o qual pode ser uma instância da classe motores e um elemento do conjunto das máquinasPotentes.

Nós percebemos, portanto, que a primeira asserção feita sobre as entidades do mundo real na seção 1, ou seja, de que tudo é um objeto do modelo, ainda está sendo satisfeita, a despeito da consideração independente dos conceitos de abstração (veja a seção 1). Entretanto, o segundo axioma só pode ser satisfeito se todos os conceitos de abstração forem considerados conjuntamente, tal que toda informação sobre uma entidade esteja concentrada em um objeto do modelo. Isto tem uma vantagem adicional. Quando se considera todos os conceitos de abstração juntos, descrições de um conceito podem usar as descrições dos outros. Isto permitirá descrições mais ricas e mais precisas das entidades do mundo que serão obtidas seguindo o desenvolvimento de modelos que reflitam mais diretamente e naturalmente a conceitualização do mundo real por parte dos usuários. Uma vantagem final é que os objetos podem ser concisamente descritos sem perder a necessária flexibilidade, e alguns erros clássicos provocados pela introdução de redundância podem ser evitados pela redução da quantidade de repetições nas descrições.

Assim, objetos do modelo de conhecimento têm agora seis tipos de relacionamento com outros objetos (instância-de, subclasse-de, elemento-de, subconjunto-de, elemento-parte-de, sub-componente-de) correspondendo aos diferentes papéis que eles desempenham em cada hierarquia de abstração. Conseqüentemente, o significado semântico dos objetos pode ser muito diferente de contexto para contexto, nos quais eles são encontrados. Há objetos que expressam instâncias em um contexto e elementos em outro, e mesmo objetos que representam classes, conjuntos e componentes

em diferentes contextos. Por exemplo, o objeto automóveis pode ser a subclasse de veículos em um contexto, onde propriedades estruturais que serão herdadas pelas instâncias são descritas. Em outro contexto, ele poderia ser o conjunto onde propriedades como carroMaisBarato, carroMaisCaro, preçoMédio e númeroDeCarros são expressas. E em um último contexto, ele poderia representar a agregação dos objetos motor, rodas e carroceria, como descrito na seção 3.1.3. Assim, o significado semântico de um objeto pode ser determinado apenas em relação a um contexto particular (isto é, juntamente com o relacionamento de abstração correspondente). A figura 14 mostra uma explanação esquemática desta idéia. A descrição completa do objeto agora incorpora tanto a classificação/generalização quanto as propriedades de associação e agregação.



1 é-instância-de
2 é-subclasse-de
3 tem-instâncias
4 tem-subclasses
5 é-elemento-parte-de
6 tem-subconjuntos
7 tem-elementos
8 é-subconjunto-de
9 é-elemento-de
10 tem-elementos-partes
11 é-sub-componente-de
12 tem-sub-componentes

Observação: A figura deve ser interpretada da seguinte maneira: OBJETO tem o significado de, por exemplo, instância, quando ele tem o relacionamento "é-instância-de" com outro objeto.

Figura 14: Resumo esquemático das abstrações semânticas

6.5 Enriquecimento Passo a Passo de Descrições de Objetos

Vamos, primeiramente, considerar as propriedades do conceito de classificação/generalização juntamente com a associação.

Quando se concentra propriedades de classes e propriedades de conjuntos em apenas um objeto (por exemplo, automóveis), as propriedades dos conjuntos serão herdadas juntamente com as propriedades das classes por todas as instâncias. Isto, certamente, é indesejável.

Para evitar este problema, é importante diferenciar propriedades de conjuntos e classes para restringir a herança às propriedades de classes. Isto é obtido pela inclusão de um tipo de propriedade

(instância ou própria) associado a cada propriedade. Desta forma, propriedades-instância são aquelas que concernem à classificação/generalização e propriedades-próprias são aquelas relacionadas à associação. Propriedades-instância são, portanto herdadas já que elas descrevem características das instâncias de uma classe, enquanto que propriedades-próprias não são herdadas já que elas descrevem características dos próprios objetos conjunto.

Naturalmente, no caso de que um objeto desempenha o papel de uma classe em um contexto e de uma instância em outro contexto, é também importante evitar a herança de propriedades de instâncias. Com certeza, propriedades que são herdadas pelo objeto no segundo contexto, já que ele é uma instância não deveriam ser mais herdadas no primeiro contexto onde ele é uma classe, já que elas descrevem características do objeto relacionadas apenas a sua existência como uma instância. Por causa disso, nós generalizamos o significado das propriedades-próprias fazendo com que não se restrinjam ao conceito de associação. Além disso, propriedades-próprias expressam propriedades próprias de qualquer objeto (como uma instância ou um conjunto). Assim, a herança ocorre sobre os relacionamentos de generalização/classificação de acordo com as seguintes regras:

a) propriedades-instância são herdadas como propriedades-instância pelas subclasses de uma classe e como propriedades-próprias pelas suas instâncias, e

b) propriedades próprias não são nunca herdadas.

Quando se combina agregação com os outros conceitos, também é importante diferenciar as propriedades da agregação de outras propriedades dos objetos (como foi feito no caso de associação e generalização/classificação) já que valores de propriedades de agregação devem ser interpretados como outros objetos do modelo. Note que a semântica das propriedades de agregação é bem diferente da semântica de propriedades de classes e conjuntos. Enquanto as propriedades de agregação representam partes dos objetos, propriedades de classes e conjuntos representam características dos mesmos (veja a seção 3.2). Como conseqüência, nós introduzimos uma indicação sobre se uma propriedade descreve um relacionamento de agregação ou não, na descrição de cada propriedade do objeto, em adição ao tipo próprio ou instância. Nós indicamos as propriedades parte-de como as não-terminais já que seus valores correspondem a outros objetos do modelo, enquanto as outras (propriedades de generalização/classificação e associação) são denotadas como propriedades terminais.

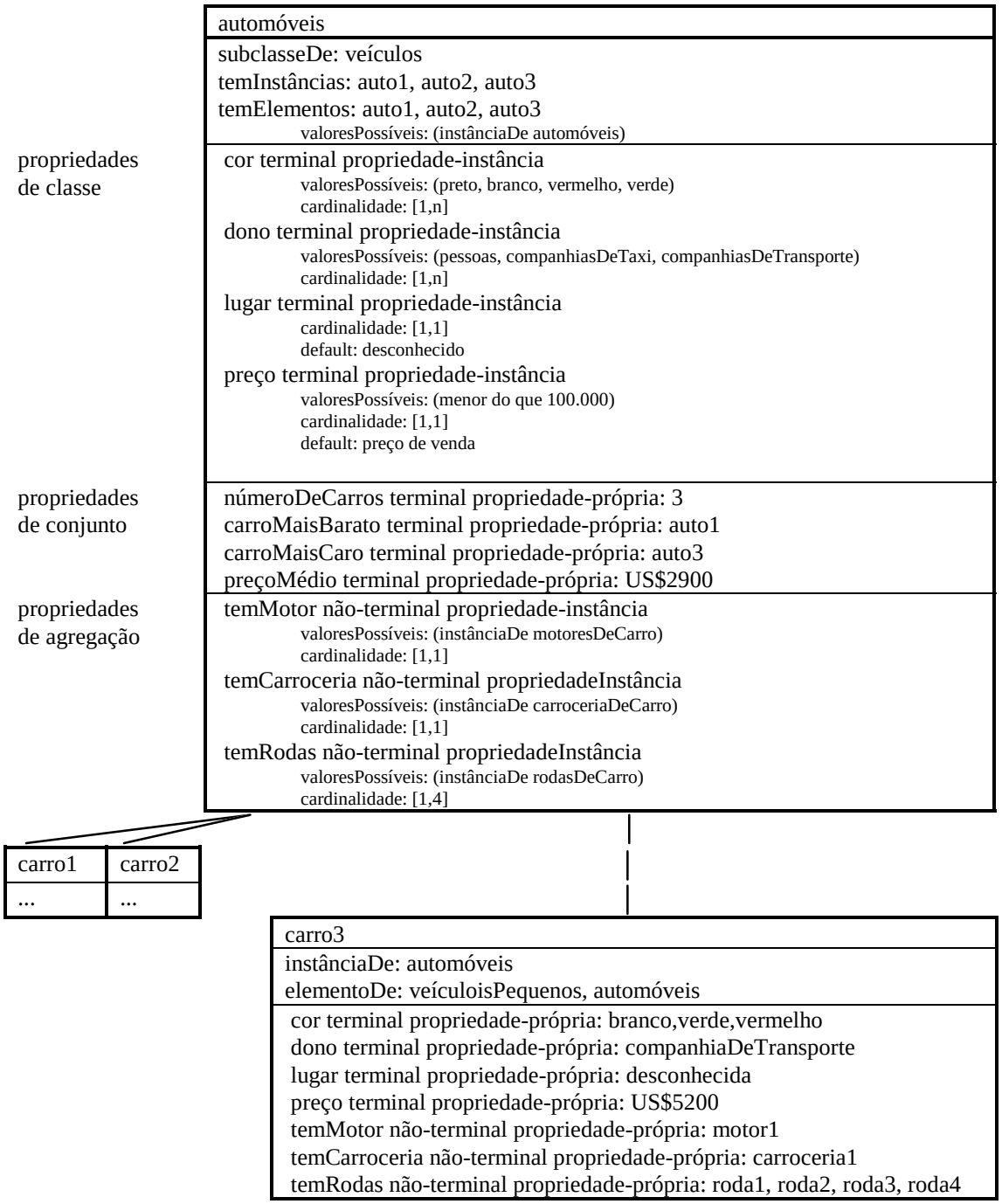


Figura 15: Uma visão integrada do objeto automóveis.

Considerando o objeto automóveis, ele tem as propriedades terminais cor, dono, lugar, preço, númeroDeCarros, carroMaisBarato, carroMaisCaro e preçoMédio, as quais representam características ou das instâncias ou do conjunto de todos os automóveis, e as propriedades não terminais, motor, rodas e carroceria, as quais representam as partes dos automóveis (figura 15). Já que cada instância de automóveis também tem as partes motor, rodas e carroceria, nós usamos o conceito de classificação e definimos estas propriedades não-terminais adicionalmente como propriedades-instâncias tal que elas são passadas como propriedades-próprias não-terminais para cada instância de automóveis. Desta maneira, nós integramos todos os conceitos de abstração e usamos as facilidades de algumas delas (no exemplo, herança) para a descrição das outras (neste

caso, agregação). O resultado, como já foi mencionado, é uma descrição abreviada. Se não for usada classificação, a definição das propriedades de agregação motor, rodas e carroceria teria que ser repetida para cada instância de automóveis.

Como foi discutido na seção 3.2. os relacionamentos de abstração devem ser explicitamente representados no modelo. Já que classes e propriedades de conjuntos não contém a informação necessária para representar os correspondentes relacionamentos de abstração, nós introduzimos algumas propriedades pré-definidas em cada objeto com o objetivo de expressar diretamente estes relacionamentos. Adicionalmente, nós combinamos estas propriedades em pares, tal que uma modelagem simétrica é obtida. A classificação é representada pelas propriedades temInstâncias, especificada em uma classe, e sua "retro-referência" instânciaDe, especificada nas instâncias correspondentes. A propriedade temSubclasses definida para uma superclasse corresponde à propriedade subclasseDe definida para uma subclasse. Juntamente, elas representam o conceito de generalização. Da mesma maneira, associação-elemento é representada pelas propriedades tem-elementos e elemento-de, enquanto que o conceito de associação-conjunto é suportado por temSubconjuntos e subconjuntoDe (veja a figura 14).

Para representar os relacionamentos do conceito de agregação, nós poderíamos usar também propriedades pré-definidas (por exemplo, subcomponente-de, tem-subcomponentes, elemento-parte-de, e tem-elementos-parte). Entretanto, isto não é necessário, já que propriedades de agregação já especificam exatamente o significado de tais relacionamentos. Além disso, quando se expressa cada relacionamento em uma propriedade, a agregação pode ser melhor combinada com outros conceitos, permitindo que restrições específicas sejam atribuídas a cada relacionamento particular, como mostrado na figura 15. Note que não seria possível ter um valorPossível e uma especificação de cardinalidade para cada subcomponente de automóveis se estes subcomponentes fossem representados juntos por apenas uma propriedade pré-definida (tem-subcomponentes) como foi feito nos outros conceitos de abstração. É claro que uma modelagem simétrica também é desejada aqui. Por causa disso, a especificação de uma propriedade representando uma "retro-referência" a ser mantida no subcomponente ou elemento é requerida quando uma propriedade não-terminal for definida.

Antes de terminar nossa discussão, nós gostaríamos de considerar um exemplo final de um modelo que integra todos os conceitos de abstração. O exemplo, mostrado na figura 16 descreve a produção de uma fábrica de automóveis hipotética. Ela produz diferentes automóveis (modelo x, modelo y,...) cada uma dos quais tem pequenas variantes (padrões, luxo,...). Automóveis, modelos e variantes são relacionados pelo conceito de generalização tal que as características gerais de automóveis são herdadas por cada modelo de automóvel, onde elas são correspondentemente especializadas e complementadas com outras características específicas a este modelo. As propriedades dos modelos são, então, herdadas por suas variantes, nas quais elas são especializadas e complementadas mais uma vez, expressando exatamente as características de cada variante de automóvel. Além disso, automóveis são definidos como uma agregação de várias partes de automóveis (baterias, motores, etc,...) as quais podem, por sua vez, consistir de outros componentes. Já que cada parte de automóvel é construída para um modelo particular, estas partes também constituem uma hierarquia é-um.

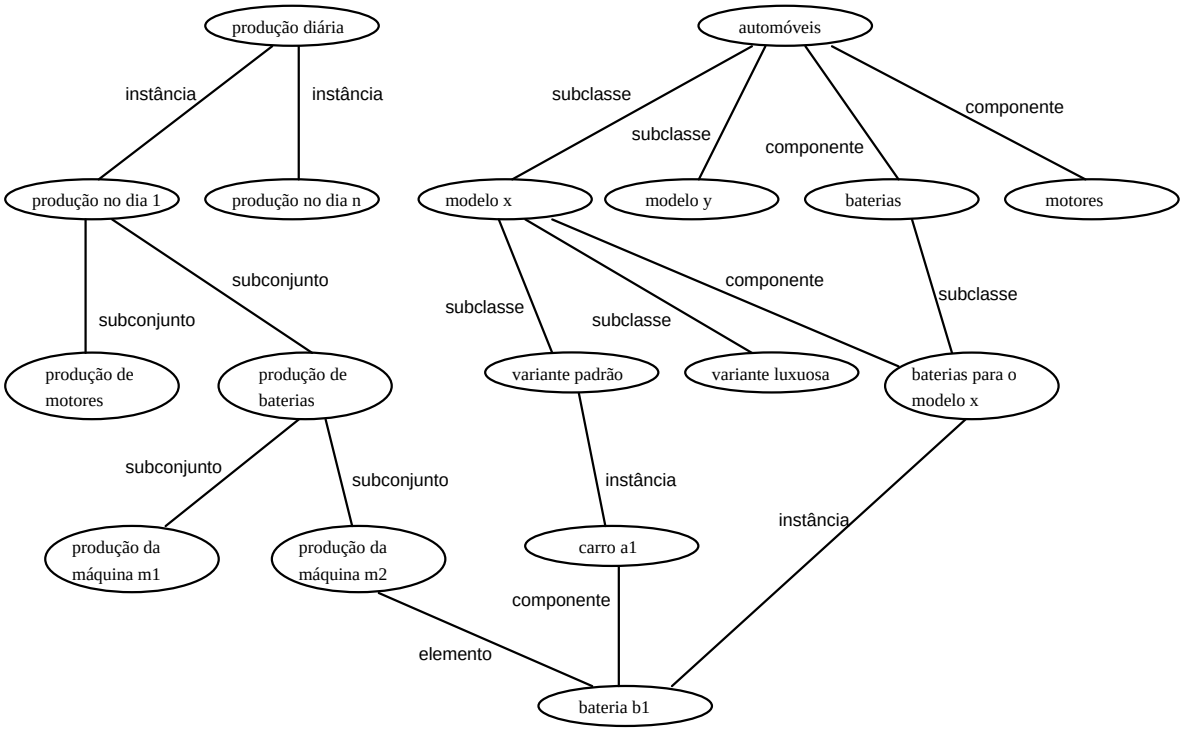


Figura 16: Exemplo de aplicação integrando os conceitos da abstração.

Nas folhas das hierarquias é-um estão os automóveis realmente produzidos (carro A1), e as partes de automóveis (bateria B1), que estão, mais uma vez, como suas superclasses, relacionadas ao conceito de agregação. Com a necessidade de um controle diário da produção para cada máquina, as partes de automóveis (bateria B1) são agrupadas em conjuntos, cujos elementos são as partes produzidas por uma máquina particular em um dia. Isto permite a comparação de performance de diferentes máquinas bem como a freqüência de produção de peças defeituosas. A produção das máquinas é agrupada em conjuntos de produção de partes (produção de baterias, produção de motores, ...) e estes, por sua vez, em um conjunto de produção diária (D1). Cada produção diária tem propriedades de conjunto como o número total de peças produzidas, número de máquinas envolvidas, etc. Com o propósito de evitar a definição destas propriedades em cada conjunto particular diário, usamos os conceito de classificação e as descrevemos como propriedades-instância na classe produçãoDiária. Sobre os relacionamentos instânciaDe, eles são herdados como propriedades-próprias, definido a estrutura (isto é, o conjunto de propriedades) de cada conjunto de produção diária.

6.6 Conclusões

Nesta seção, nós discutimos os conceitos de abstração: as construções mais importantes a ser suportadas pelo modelo de conhecimento fornecido pelo nível de engenharia. Nós preferimos considerá-las de um ponto de vista neutro, tratando-as da mesma forma que observadores humanos consideram a conceitualização do mundo real, ao invés de fornecer uma descrição detalhada da realização em um modelo particular. O foco desta seção esteve, por esta razão, principalmente na definição da semântica delas, em mostrar sua aplicabilidade e em apresentar um modo de integrá-las ao modelo de conhecimento.

Nós sustentamos que descrições de objetos devem incorporar todos os conceitos de abstração, tal que os objetos possam desempenhar diferentes papéis (isto é, classe, conjunto,

componente, etc) ao mesmo tempo, dependendo dos relacionamentos que eles têm com outros objetos. Tendo em vista a existência destes relacionamentos, a mais importante vantagem dos conceitos de abstração (suas facilidades de raciocínio embutidas) é usada para fazer deduções sobre objetos:

a) herança é usada para deduzir a estrutura interna das instâncias,

b) condições de pertinência são usadas para deduzir crenças sobre elementos definindo diferentes "visões" dos mesmos.

c) predicados implicados são usados para tirar conclusões sobre objetos agregação, baseados na monotonia das propriedades.