

Uma Representação Gráfica para o Modelo de Objetos do GEOTABA

Luiz Carlos Montez Monte

COPPE - SISTEMAS / UFRJ

Caixa Postal 68511

CEP: 21.945 - Rio de Janeiro - RJ

phones: (021)590-2552, (021)590-2652, (021)290-4540

fax: (021)290-6626

e-mail: COS10002@ufrj.bitnet

Resumo - O Modelo de Objetos do GEOTABA está sendo definido de modo a poder ser implementado diretamente pelo sistema de gerência de objetos GEOTABA, apesar de possuir um alto nível que permitirá seu uso na modelagem de informação e sistemas. Os Diagramas de Objetos são uma notação gráfica usável na modelagem conceitual de bases de objetos do GEOTABA, representando visualmente os conceitos contidos no seu Modelo de Objetos. Ao mesmo tempo que os Diagramas de Objetos sejam adequados à representação de modelagens semântica da informação, eles devem ser mapeados trivialmente no Modelo de Objetos do GEOTABA. Este trabalho apresenta os Diagramas de Objetos, mostrando, através de um exemplo, como a semântica da informação contida em uma modelagem tradicional, utilizando um Diagrama de Entidades-Relacionamentos comum, pode ser representada pelos Diagramas de Objetos. É mostrado também como as noções introduzidas neste exemplo são capazes de modelar propriedades não representáveis pelos Diagramas de Entidades-Relacionamentos convencionais. Extensões aos Diagramas de Objetos estão previstas, de modo a incorporá-los a uma linguagem visual de definição e manipulação de objetos para o GEOTABA. Os Diagramas de Objetos têm sido empregados na especificação do metaesquema do GEOTABA e como ferramenta para ajudar a compreensão e o aperfeiçoamento do Modelo de Objetos do GEOTABA.

1. Introdução

O sistema de gerência de objetos GEOTABA, em desenvolvimento no Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, pode ser classificado como um banco de dados orientado a objetos comportamentalmente. A informação por ele gerenciada é organizada de acordo com um Modelo de Objetos [MONTE 90] que engloba regras de definição de estruturas e de procedimentos de manipulação da informação. O Modelo de Objetos do GEOTABA, conforme proposto, é uma ferramenta de alto nível semântico, capaz de ser usada diretamente na modelagem de sistemas e da informação por eles manipulada. Apesar disso, este modelo é implementado diretamente pelo sistema de gerência de objetos GEOTABA. Este atendimento simultâneo ao homem e à máquina deve privilegiar os fatores humanos, correspondendo à tendência atual.

A utilização intensiva da capacidade gráfica dos computadores tem sido carta fundamental para o aumento da aceitação dos sistemas pelos indivíduos. A criação de uma representação gráfica para o Modelo de Objetos é uma providência que visa incrementar a usabilidade deste modelo.

Objetivo - Este trabalho apresenta a notação DO - Diagrama de Objetos -, representação gráfica do Modelo de Objetos do sistema de gerência de objetos GEOTABA. Esta representação gráfica é usável na modelagem conceitual de bases de objetos do GEOTABA, seja esta realizada manualmente ou com auxílio de computador.

A função principal do DO pode ser comparada à função da representação gráfica DER - Diagramas de Entidades-Relacionamentos. Ambas pretendem poder representar modelagens semânticas de informação. Todavia, sem qualquer prejuízo quanto à capacidade de representar estruturas e restrições, sem impor mecanismos extravagantes que dificultem a modelagem ou a torne artificial demais, o DO tem que se manter diretamente mapeável no Modelo de Objetos do GEOTABA. Assim, um diagrama deve corresponder exatamente a um esquema de classes e objetos implementável no GEOTABA.

A representação gráfica proposta deve ser adequada a utilização a mão livre, não contendo símbolos difíceis de desenhar. A possibilidade de se ter ferramentas de software para auxílio a confecção desses diagramas permite encarar essa representação gráfica como uma verdadeira linguagem visual. Para tal, os DOs devem poder representar totalmente o Modelo de Objetos, não apenas o nível conceitual dele. Para isto ser possível, os diagramas devem ser estendidos com mecanismos para representação de outros níveis de abstração do modelo: lógico, externo e apresentação. A possibilidade de se representar métodos e de se executar interativamente mensagens aos objetos, completarão a linguagem visual.

Escopo do Trabalho - Este relatório se concentra na descrição das características dos Diagramas de Objetos comparáveis com as características dos Diagramas de Entidades-Relacionamentos tradicionais. Outras possibilidades dos Diagramas de Objetos, capazes de representar outros tipos de restrições de integridade e características do Modelo de Objetos não capturáveis pelos Diagramas de Entidades-Relacionamentos convencionais, não são discutidas aqui. O trabalho busca mostrar como toda a informação modelável através de um DER pode ser representada por um DO.

Embora os Diagramas de Objetos possam colaborar para uma melhor compreensão do Modelo de Objetos do GEOTABA, este não é descrito neste trabalho. Apenas os conceitos relevantes do Modelo de Objetos, referentes a comparação entre os dois tipos de diagramas, são aqui comentados. Para um pouco mais de informação sobre o Modelo de Objetos, é possível referir-se a [MONTE 90].

Trabalhos Correlatos - Notações gráficas para representar modelagens orientadas a objetos têm sido propostas e discutidas, ainda que escassamente [BOOCH 91, COAD 90, HENDERSON 90, ALABISO 88]. Modelos de dados semânticos produziram várias notações gráficas para modelagem conceitual, por exemplo, GSM [HULL 87], DER [CHEN 76], EER

[TEOREY 86], FDM [SHIPMAN 81, DAYAL 84]. O aproveitamento de qualquer destas notações é impedido pelas características específicas do Modelo de Objetos do GEOTABA, tais como: distinção entre classes e coleções (incomum nos modelos semânticos); associações constantes e variáveis; definição de relacionamentos (inexistentes nas notações orientadas a objetos) através de mapeamentos; e herança e sobreposição de propriedades, inclusive de mapeamentos.

Organização do Artigo - Os Diagramas de Objetos são ferramentas para modelar conceitualmente informação de acordo com características do Modelo de Objetos do GEOTABA. Portanto, certas peculiaridades do Modelo de Objetos, referentes ao nível conceitual de abstração, impõem requisitos sobre a representação gráfica, isto é, sobre os diagramas. Outros requisitos atendidos pelos Diagramas de Objetos se referem a características desejáveis de manipulação dos diagramas pelo usuário. A seção 2 enumera características relevantes do Modelo de Objetos e características de utilização pelo usuário e os requisitos correspondentes.

A seção 3 apresenta as noções envolvidas nos Diagramas de Objetos através da modelagem conceitual da informação de um exemplo típico utilizado na literatura para estabelecer as propriedades dos Diagramas de Entidades-Relacionamentos. Finalizando, na seção 4, são apresentados um sumário, perspectivas futuras e conclusões sobre o apresentado.

2. O Modelo de Objetos do GEOTABA

O Modelo de Objetos do GEOTABA é uma ferramenta conceitual equivalente a uma linguagem com semântica definida, porém sem uma sintaxe particular. Para o mesmo modelo de objetos, diferentes sintaxes podem ser estabelecidas. A notação gráfica proposta pode ser vista como uma possível sintaxe para simbolizar alguns conceitos do Modelo de Objetos.

Os Diagramas de Objetos aqui apresentados são uma sublinguagem dessa linguagem gráfica. Apenas a parte declarativa de definição de objetos é contemplada neles. A possibilidade dos diagramas poderem representar esta parte do Modelo de Objetos impõe requisitos específicos ao projeto destes diagramas. A seguir estão enumerados alguns dos requisitos para a representação gráfica relativos ao Modelo de Objetos.

Classes e Coleções - O Modelo de Objetos separa o conceito de classe de objetos do conceito de coleções de objetos. Classes reúnem apenas descrições de propriedades comuns a determinados objetos. Uma coleção é um depósito de objetos. As consultas, em geral, se referem às coleções.

O diagrama deve permitir a representação da definição de classes e, coletivamente, de esquemas de classes. O diagrama, além disso, deve permitir a definição de objetos globais persistentes, em geral coleções que formam bases de objetos, e, coletivamente, de esquemas de objetos. O diagrama deve poder ser visto em abstrações diferentes. Pode-se usá-lo tanto para representar um esquema de classes, como para representar um esquema de objetos, ou ainda para representar simultaneamente os dois esquemas.

Constantes e Variáveis - O Modelo de Objetos utiliza associações de símbolos com os objetos para que os usuários possam manipular os objetos através de nomes com significado adequado. O diagrama deve poder representar símbolos como nomes ou expressões em português.

Uma associação (símbolo, objeto) que, uma vez criada, não é mais mudada, isto é, o objeto é sempre o mesmo, é chamada de uma associação constante. O objeto envolvido em uma associação constante pode ser modificado internamente, contanto que continue sendo sempre o mesmo objeto (o mesmo identificador intrínseco de objetos). Isto é o que ocorre comumente com as coleções que formam uma base de objetos. Cada coleção tem um nome sempre associado a ela, mas a coleção em si pode variar. Um objeto que não varia seria um objeto constante.

Uma associação cujo símbolo pode, em instantes diferentes, estar associado a objetos diferentes é chamada de associação variável ou simplesmente de variável. Os diagramas devem permitir a representação distinta de associações constantes e de variáveis.

Relacionamentos - As associações internas de um objeto, denominadas variáveis de instância, implementam relacionamentos entre objetos. No nível conceitual, estes relacionamentos são classificados em atributos do objeto, componentes e mapeamentos. Os Diagramas de Objetos devem representar distintamente os diversos tipos de relacionamento. Todavia, a semelhança entre os diversos tipos de relacionamento deve ser notável nas suas representações gráficas.

A principal semelhança é que um relacionamento binário define um mapeamento de um objeto em outros. Este relacionamento pode definir um outro mapeamento, inverso do anterior. Neste caso, os dois mapeamentos formam um único relacionamento e o vínculo entre eles, nos diagramas, deve ser evidente. Descrevendo um relacionamento entre objetos, em cada objeto envolvido, através de atributos representando os seus mapeamentos, permite-se ter uma visão orientada a objetos dos relacionamentos.

Relacionamentos não binários e relacionamentos com atributos não têm suporte direto no Modelo de Objetos. Eles são modelados como objetos quaisquer. Nos diagramas, a representação destes relacionamentos deverá se aproximar mais das representações de objetos comuns do que de relacionamentos binários.

Cardinalidade dos Mapeamentos - Também de acordo com o modelo, os mapeamentos têm nome e cardinalidade mínima e máxima. Assim, um mapeamento pode ser opcional ou obrigatório e ser univariado ou multivariado. Estas diferenças devem ser explícitas nos diagramas. Um mapeamento opcional significa que o objeto pode ser associado ao objeto nulo. Um mapeamento multivariado, de acordo com o Modelo de Objetos, é o relacionamento do objeto com uma coleção de objetos e assim ele deve ser representado no diagrama.

Domínios Predefinidos - Por último, os mapeamentos podem ser feitos para objetos definidos na modelagem ou para objetos de domínios predefinidos do sistema. Os diagramas devem ter formas diferentes para representar mapeamentos em domínios triviais predefinidos pelo sistema, mapeamentos em objetos definidos no próprio diagrama e mapeamentos em objetos que não sejam do sistema, mas já tenham sido definidos. Este último tipo de representação de mapeamentos é usável no desenvolvimento incremental dos esquemas.

Serviços - Além dos atributos, componentes e mapeamentos, um objeto pode definir serviços, ou seja, processamento. Estes devem poder ser representados nos diagramas.

Classes e Herança - No Modelo de Objetos, a classificação relaciona um objeto à sua classe. Deve ser simples a identificação das classes dos objetos representados em um diagrama. A herança de propriedades de uma superclasse pelas suas subclasses deve poder ser representada nos diagramas. Do mesmo modo deve ser possível se representar a redefinição de propriedades em uma subclasse. As pseudoclasses são classes que não têm instâncias, são usadas apenas para serem superclasses de outras classes. As pseudoclasses devem ser representadas distintamente das demais classes.

Chaves - O Modelo de Objetos do GEOTABA suporta a definição de restrições de integridade sobre coleções de objetos. Por exemplo, um determinado atributo dos elementos de uma dada coleção pode ser uma chave para esta coleção, ou seja, não aceitar valores nulos, nem ter dois elementos na coleção com valores iguais nesse atributo. Os diagramas devem prover mecanismos de definição de quais atributos são chaves em uma coleção.

Nomes Implícitos - O modelo é bastante flexível quanto a necessidade de se dar nomes às entidades modeladas. Assim, pode-se ter classes sem nomes e o mesmo nome poder identificar entidades de tipos diferentes: classes, coleções, atributos, mensagens, etc. A representação gráfica deve simplificar ao máximo a tarefa do seu usuário de escolher nomes para as coisas. Isto pode ser alcançado definindo-se, na representação, regras para a definição de nomes, isto é, na ausência de uma especificação de nome feita pelo usuário. Entretanto, o usuário poderá definir nomes adequados quando bem desejar.

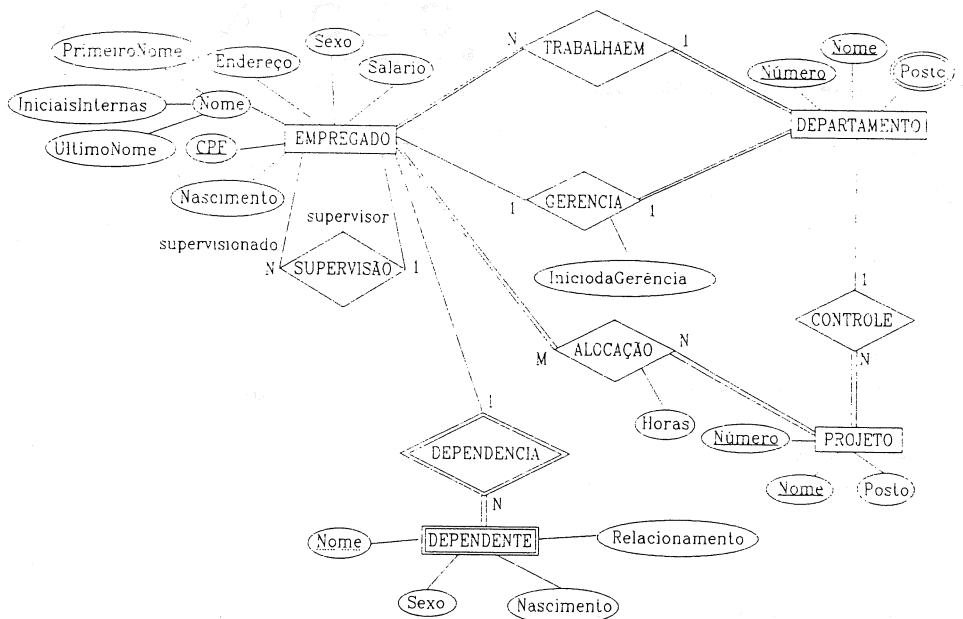


Figura 1: Diagrama Entidade-Relacionamento para a Companhia

Níveis de Visualização - Outros requisitos podem ser colocados independentemente do Modelo de Objetos. Por exemplo, a representação gráfica deve permitir múltiplos níveis de abstração. Uma mesma modelagem realizada com esses diagramas pode ter várias maneiras de ser apresentada, dependendo do tipo de informação que se deseja visualizar.

Os diagramas não devem obrigar o usuário a fazer declarações e definições que possam ser inferidas. Desse modo, a representação gráfica deve ser lacônica, sem prejuízo para a sua expressividade.

3. Um Exemplo de Modelagem

Este trabalho enfoca soluções contidas nos Diagramas de Objetos para que possam ser utilizados na modelagem conceitual da estruturas dos objetos de uma base. Sob esse aspecto, os Diagramas de Objetos podem ser comparados aos Diagramas de Entidades-Relacionamentos. Todavia, os Diagramas de Objetos devem representar o mais diretamente possível os esquemas implementáveis pelo Modelo de Objetos do sistema de gerência de objetos GEOTABA.

As características dos Diagramas de Objetos serão mostradas através de um exemplo. Visando a comparação com os Diagramas de Entidades-Relacionamentos, o exemplo escolhido é um já utilizado para exemplificar estes diagramas [ELMASRI 89]. O

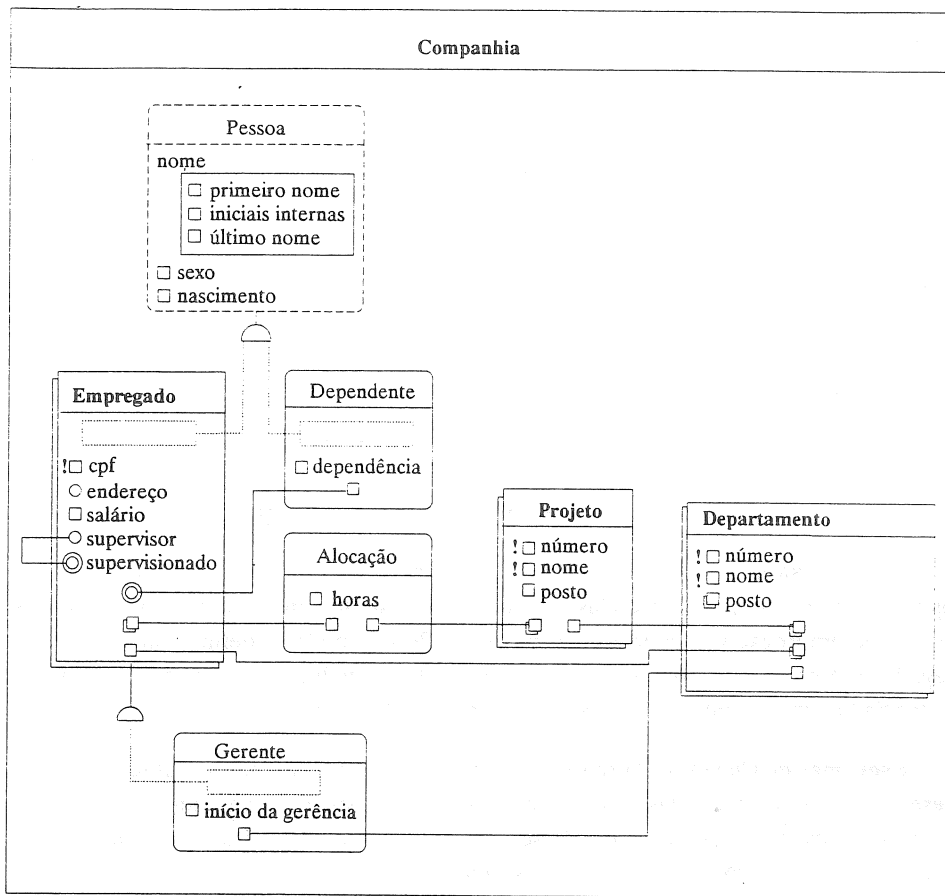


Figura 2: Diagrama de Objetos para a Companhia

exemplo descreve uma base de dados COMPANHIA, através do Diagrama Entidade-Relacionamento da Figura 1, que armazena informação sobre os empregados, departamentos e projetos de uma companhia. Os seguintes requisitos são representados no Diagrama Entidade-Relacionamento:

A Figura 2 representa, com um Diagrama de Objetos, o esquema de objetos e classes para uma base de objetos Companhia, de acordo com o Modelo de Objetos do GEOTABA. Através desse exemplo, serão mostradas, a seguir, as soluções empregadas pelos Diagramas de Objetos para atender aos requisitos estabelecidos anteriormente.

Notações de Classes e Coleções - O Diagrama de Objetos da Figura 2 representa simultaneamente os esquemas de classes e de objetos. As entidades Empregado, Projeto e Departamento estão representadas como coleções globais de objetos, contrastando com

Pessoa, Dependente, Alocação e Gerente que são classes cujos objetos não estão

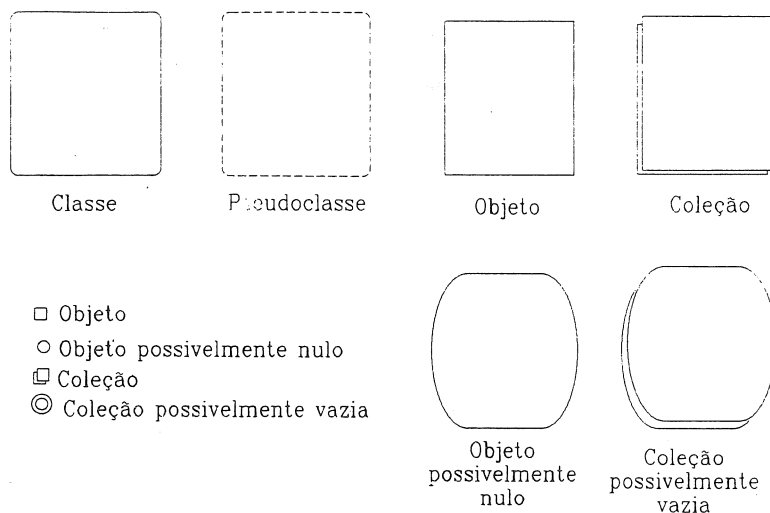


Figura 3: Ícones e Caixas para Classes e Objetos

reunidos em uma única coleção. A Figura 3 mostra as notações diferentes de classes e objetos. Com a distinção entre classes e coleções pode-se representar as propriedades comuns de empregados e dependentes na classe Pessoa, pode-se definir que certos empregados são gerentes e pode-se dispensar uma coleção global de dependentes. Assim, a existência de um dependente depende da sua associação a um empregado.

Esquemas de Classes e Coleções - A representação gráfica pode ser usada para representar o esquema de classes e o esquema de coleções em diagramas separados. A Figura 4 apresenta o esquema de classes correspondente ao diagrama da Figura 2. O esquema de objetos correspondente é mostrado na Figura 5.

Confrontando-se as três figuras, conclui-se que, no diagrama geral da Figura 2, as três coleções, Empregado, Projeto e Departamento, definem implicitamente três classes correspondentes. Na verdade, neste diagrama, a notação das coleções com classes implicitamente definidas é uma abreviatura, conforme mostrado na Figura 6.

O Diagrama de Objetos da Figura 2 é um esquema que pode ser feito para a Companhia. De acordo com o método de modelagem adotado, outros esquemas poderiam ser criados para representar a mesma informação. Por exemplo, não há necessidade de se ter três coleções globais (Empregado, Projeto e Departamento), bastando existir uma delas. Como todos os objetos estarão ligados entre si, uma coleção persistente garante a persistência dos demais objetos.

Todavia, se só existir uma das coleções globais, todo acesso a qualquer informação será feito através desta coleção. Na modelagem adotada foi empregado um outro critério: toda entidade forte corresponde a uma coleção global. Com isto se tem acesso aos

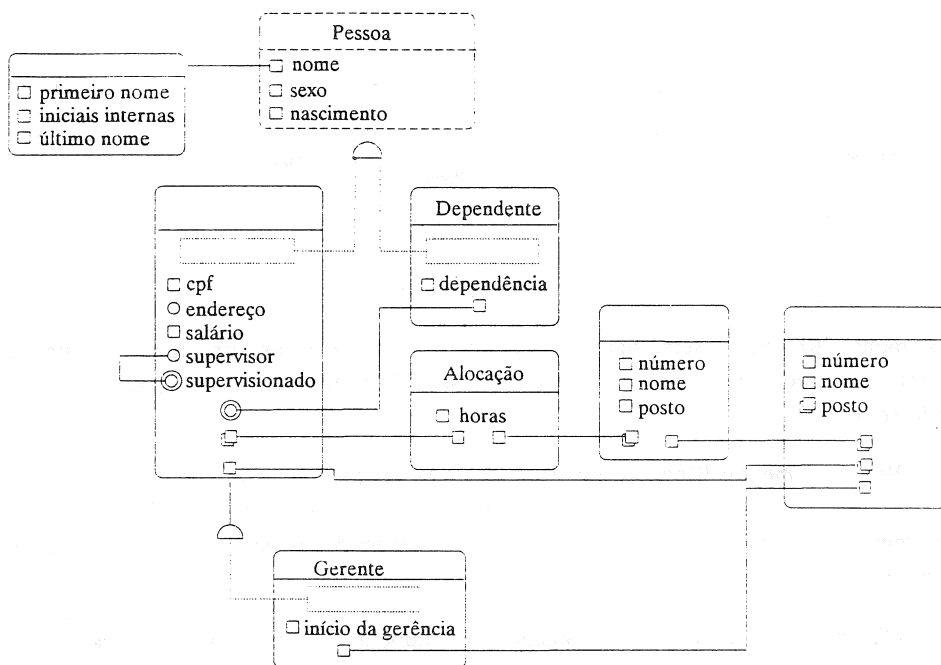


Figura 4: Esquema de Classes

empregados da companhia independentemente dos projetos e departamentos; se tem acesso aos projetos independentemente, e aos empregados também.

Representação de Associações Constantes e Variáveis - Os nomes Empregado, Projeto e Departamento, dentro do escopo da base de objetos da Companhia, estão associados a coleções. Estas são associações constantes. Isto significa que a coleção Empregado pode ter o seu conteúdo alterado, colocando-se ou retirando-se empregados, porém o nome Empregado estará constantemente associado a esta mesma coleção. Outras associações constantes envolvem os nomes das classes Pessoa, Dependente, Alocação e Gerente.

Já os atributos foram todos modelados usando-se associações variáveis. Por exemplo, todo empregado tem um salário, porém este pode ser alterado. A regra para representar associações variáveis e constantes nos Diagramas de Objetos é a seguinte: uma associação constante de um nome a uma entidade é feita colocando-se o nome dentro da representação da entidade; em uma associação variável, o nome fica fora da representação da entidade.

Alguns atributos poderiam ser modelados usando associações constantes, como na Figura 7. Neste diagrama, não é permitido que um empregado mude o seu CPF. Também não é permitido que uma pessoa na companhia mude de sexo ou de data de

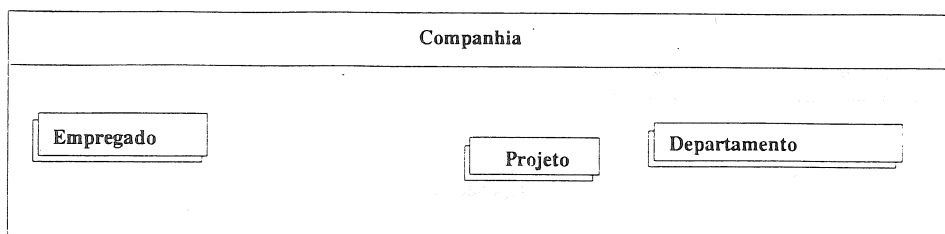


Figura 5: Esquema de Objetos

nascimento. Restrições semelhantes poderiam ser colocadas caso um departamento ou projeto não pudessem mudar de número ou se a data de início de uma gerência não pudesse ser alterada.

Relacionamentos Binários sem Atributos - As ligações entre as entidades representam os tipos de relacionamentos existentes entre elas. Assim vê-se, na Figura 2, que empregados e departamentos se relacionam. As extremidades das ligações representam os mapeamentos.

Observando-se a ligação Empregado-Departamento, interpreta-se que, através desta ligação, um empregado tem um atributo que representa um mapeamento em um departamento. Implicitamente, o nome deste mapeamento é "departamento". Portanto, dado um empregado pode-se falar no "departamento do empregado".

O mapeamento inverso também está especificado. Um departamento tem um atributo, com o nome "empregado" implícito, que representa uma coleção de empregados. Um objeto percebe o seu relacionamento com outro(s) objeto(s) através de um atributo equivalente ao mapeamento do objeto no(s) outros(s). A representação gráfica adotada segue a linha de representar os relacionamentos com uma visão orientada a objetos.

Outros Relacionamentos - Como mencionado anteriormente, relacionamentos não binários ou com atributos têm que ser tratados como objetos em si. Isto é exemplificado, na Figura 2, pela ligação Empregado-Projeto através da classe Alocação. Um empregado pode trabalhar em vários projetos e um projeto pode ter vários empregados a ele alocados. Isto explica os atributos em Empregado e Projeto. Entretanto, cada alocação de um empregado a um projeto precisa guardar um atributo contendo o número de horas que o empregado está alocado a este projeto.

Modelando uma alocação como sendo um tipo de objeto, conforme a Figura 2, um empregado tem um conjunto de alocações a projetos e um projeto tem um conjunto de alocações de empregados. Neste diagrama, o relacionamento Empregado-Departamento foi partido em dois relacionamentos: Empregado-Alocação e Projeto-Alocação. O defeito desta quebra é visível quando se nota que nada restringe um empregado a ter

duas alocações para um mesmo projeto. A Figura 7 apresenta uma notação que, não partindo a ligação Empregado-Projeto, soluciona esta questão.

Mapeamento Sem Inverso -

Pela Figura 2 observa-se que empregados se relacionam com objetos da classe Dependente. A representação utilizada para o atributo de Empregado correspondente ao seu mapeamento em dependentes significa que o valor deste atributo é uma coleção possivelmente vazia de objetos da classe Dependente.

Inversamente, em Dependente há um atributo que indica de qual empregado a pessoa é dependente. Esta é uma opção da modelagem da base de objetos. Caso seja assumido que os dependentes só são manipulados a partir dos empregados, o último mapeamento não precisa ser especificado, como na Figura 8.

Nomes de Mapeamentos Explícitos - O auto-relacionamento Empregado-Empregado, indicando que um empregado pode (ou não) ter supervisor e pode (ou não) ter empregados supervisionados, é representado pelos atributos **supervisor** e **supervisionado**. Estes atributos tiveram que receber nomes explicitamente, ao contrário dos demais atributos de mapeamento. Isto foi necessário porque a regra de nomes implícitos estabelece que o nome de um mapeamento é o mesmo da classe ou objeto para o qual será feito o mapeamento. Assim, o atributo em Empregado que mapeia para os seus dependentes se chama **dependente**, pois mapeia para objetos da classe Dependente. Se este nome não fosse adequado para explicar o mapeamento, o projetista poderia especificar um outro explicitamente. Para evitar ambiguidade, o auto-relacionamento não pode usar nomes implícitos.

Classes Implícitas - Os atributos **cpf**, **endereço** e **salário** de Empregado também representam mapeamentos. Nos diagramas não foram empregadas ligações para ligar estes atributos às classes de seus elementos porque, no projeto, estas classes foram consideradas triviais e predefinidas (texto, número, data, etc.).

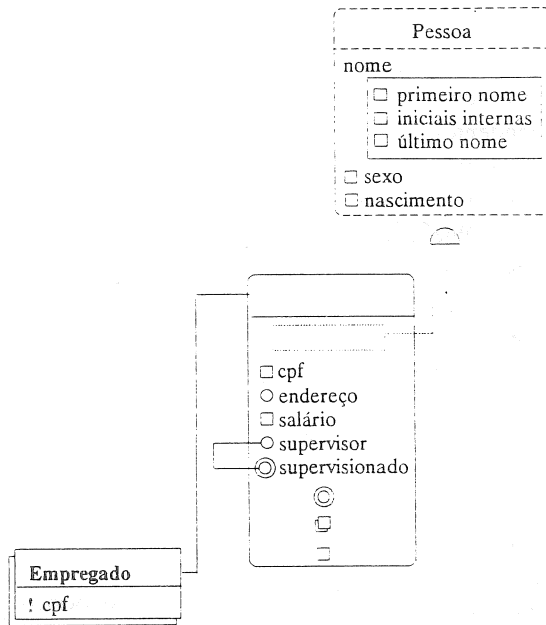


Figura 6: Expansão da Coleção Empregado

Herança e Pseudoclasses - Pelo DER da Figura 1 pode-se verificar que tanto empregados como seus dependentes possuem atributos comuns: nome, sexo e nascimento. Nos Diagramas de Objetos esses atributos poderiam ser colocados em evidência com a criação de uma superclasse para empregados e dependentes. Isto está representado na Figura 8, com a criação da classe Pessoa. Tanto Empregado como Dependente são Pessoa e herdam suas propriedades. Como não existe, nesta base, uma pessoa que não seja um empregado ou um dependente, Pessoa é uma pseudoclasse e assim foi representada. Note-se também que os nomes de dependentes passaram a ser vistos do mesmo modo que os nomes de empregados: estruturados em **primeiro nome**, **iniciais internas** e **último nome**. Em um DER convencional, sem definição de classes, tipos ou domínios, esta possibilidade aumentaria a complexidade do diagrama resultante.

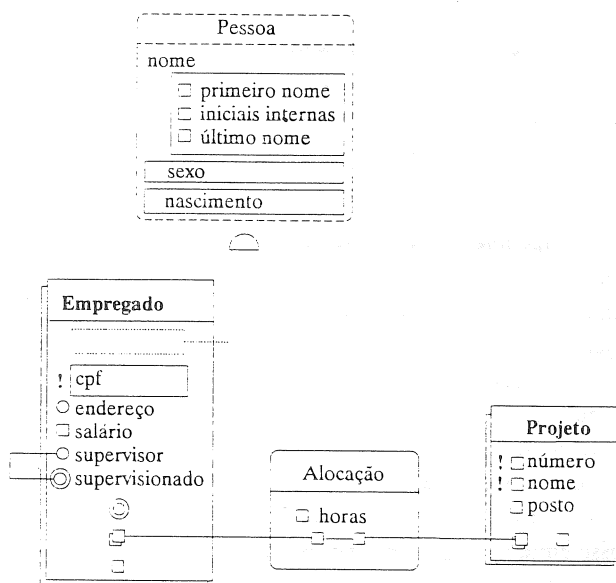


Figura 7: Associações Constantes em Atributos e Ligação Empregado-Projeto através de Alocação

A estruturação de **nome** de Pessoa é realizada dentro da própria caixinha que designa o atributo. Isto é feito assim por comodidade. A classe que descreve a estrutura dos nomes de pessoas é derivável automaticamente deste diagrama, como pode ser observado na Figura 4. Uma modelagem alternativa criaria explicitamente uma classe para nomes de pessoas e faria uma ligação do atributo **nome** de Pessoa a essa classe.

Alguns empregados gerenciam departamentos. Este relacionamento parcial fica melhor modelado com a definição da subclasse Gerente que representa os empregados envolvidos neste relacionamento. A data de início da gerência passa a ser um atributo exclusivo desse tipo de empregado.

Sobreposição - Nada impede, no DER da Figura 1, que um gerente trabalhe em um departamento mas gerencie outro. O Diagrama de Objetos apresentado proíbe esta situação usando o fato que uma subclasse pode alterar a definição de alguma propriedade que ela tenha herdado de suas superclasses. Isto é feito do seguinte modo: o relacionamento Empregado-Departamento definiu implicitamente em Empregado um atributo

departamento que indica qual departamento o empregado trabalha. Um gerente ao se relacionar com o departamento que gerencia, na ligação Gerente-Empregado, redefine o atributo **departamento**, pois não foi dado um outro nome explicitamente para este mapeamento. Portanto, o departamento de um gerente é o que está envolvido no relacionamento de gerência. Esta característica permite uma representação elegante deste tipo de restrição.

Restrições de Chave - O exemplo define chaves candidatas para as coleções Empregado (cpf), Projeto (número e nome) e Departamento (número e nome). Os Diagramas de Objetos representam os atributos chaves precedendo-os por um sinal de exclamação (!).

A restrição de que, para um mesmo empregado, não há dependentes com o mesmo nome é definida indicando que a coleção de dependentes de cada empregado, representada pelo atributo de nome implícito **dependente**, tem **nome** como chave. Entretanto, esta restrição deve ser colocada apenas na coleção referente ao atributo **dependente**, como na Figura 9.

4. Considerações Finais

Sumário - Este trabalho apresentou os Diagramas de Objetos, usáveis na modelagem conceitual de bases de objetos do GEOTABA. A propriedade fundamental dos Diagramas de Objetos é atender ao mesmo tempo a representação de modelagens semânticas de informação e serem mapeados trivialmente no Modelo de Objetos do GEOTABA.

Os diagramas espelham diversos conceitos do Modelo de Objetos, tais como: objetos, classes e coleções; associações constantes e variáveis; reconhecimento da existência de relacionamentos entre objetos e representação destes relacionamentos através de seus mapeamentos; atributos e mapeamentos opcionais e obrigatórios, univalorados e multivalorados; herança e sobreposição de propriedades através de hierarquia de classes; chaves de coleções.

O foco desta apresentação foi dado em como a semântica da informação contida em uma modelagem tradicional, utilizando um DER comum, pode ser representada pelos Diagramas de Objetos. Isto foi mostrado através de um Diagrama de Objetos que mo-

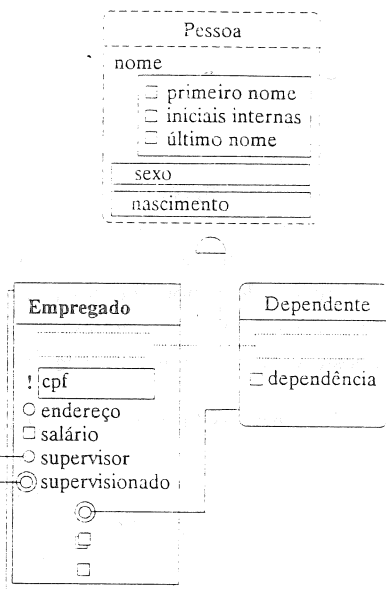


Figura 8: Mapeamento sem Inversa

dela a informação descrita em um DER utilizado didaticamente para exemplificar as possibilidades dos Diagramas de Entidades-Relacionamentos. Também foi mostrado que as noções empregadas neste Diagrama de Objetos exemplo são capazes de modelar propriedades não representáveis pelos Diagramas de Entidades-Relacionamentos tradicionais, tais como, associações constantes, herança e sobreposição de atributos e de relacionamentos.

Perspectivas Futuras - Os Diagramas de Objetos, conforme mostrados neste trabalho, servem como base para extensões que introduzam noções adequadas a capturar outros tipos de restrições diferentes das tratadas no exemplo utilizado. Especificamente, este é o caso da junção e disjunção de ligações, com a finalidade de se estabelecer condições de integridade que envolvam mais de um relacionamento.

Uma série de conceitos definidos no Modelo de Objetos do GEOTABA não foram empregados neste exemplo e, consequentemente, não tiveram suas representações introduzidas nos Diagramas de Objetos deste trabalho. Este é o caso da especificação de atributos componentes, da especificação de cardinalidade mínima e máxima pouco usuais, a utilização de objetos e classes pré-existentes ou padrões, e da especificação do protocolo de serviços realizáveis pelos objetos.

Mais além, os Diagramas de Objetos devem ser encarados como um subconjunto de uma linguagem visual de definição e manipulação de objetos do GEOTABA. Para isto ser alcançado, deve ser possível especificar interativamente processamentos e executá-los. Os passos seguintes serão fornecer a possibilidade de se representar não apenas o nível conceitual, mas também os demais níveis de abstração do Modelo de Objetos: nível externo, de representação e de implementação. Isto inclui a especificação dos mapeamentos entre os diferentes níveis de abstração.

Conclusões - Os Diagramas de Objetos têm sido empregados na especificação do metamodelo do GEOTABA e como ferramenta para ajudar a compreensão e o aperfeiçoamento do seu Modelo de Objetos. Como os conceitos deste modelo são representados graficamente nos diagramas, as pessoas encontram mais facilidade de fixar as noções envolvidas e seus propósitos.

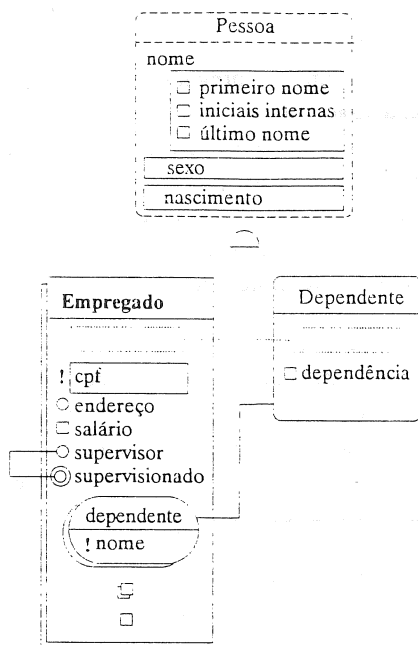


Figura 9: Chave Parcial

Como etapa preliminar para a especificação do metadesquema do GEOTABA, foi elaborado um Diagrama de Objetos representando o metadesquema do sistema Smalltalk/V [DIGITALK 86]. Por metadesquema, entenda-se o esquema de classes e objetos predefinidos envolvidos na criação e manipulação básicas de classes e objetos, ou seja, as classes **Object**, **Class**, **Metaclass**, **Behavior**, **CompiledMethod**, etc. Algumas extensões foram necessárias para se poder representar serviços prestados por estas classes (compilar um texto fonte, criar um objeto, executar uma mensagem, etc.) e para representar restrições não convencionais.

Embora não tenha sido desenvolvido um editor de Diagramas de Objetos, dois fatos permitem prever que não haverá dificuldades maiores nesta implementação. O primeiro deles é a trivialidade do mapeamento da representação gráfica nos Diagramas de Objetos em classes e objetos do GEOTABA. O segundo fato é a experiência na realização de um editor de diagramas semelhantes [MATTOSO 90].

5. Referências

- [ALABISO 88] B. ALABISO, *Transformations of Data Flow Analysis Models to Object-Oriented Design*, in *Proceedings of OOPSLA'88*, ACM, New York, pp. 335-353, 1988.
- [BOOCH 91] GRADY BOOCH, *Object-Oriented Design with Applications*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1991.
- [CHEN 76] P.P. CHEN, *The Entity-Relationship Model - Toward a Unified View of Data*. *ACM Trans. Database Syst.* 1:1, pp. 9-36, março 1976.
- [COAD 90] PETER COAD, EDWARD YOURDON, *Object-Oriented Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1990.
- [DAYAL 84] U. DAYAL, H.Y. HWANG, *View Definition and Generalization for Database Integration in a Multidatabase System*. *IEEE Trans. Soft. Eng.* SE-10:6, pp. 628-644, 1984.
- [DIGITALK 86] *Smalltalk/V: Tutorial and Programming Handbook*, Digitalk Inc., Los Angeles, Calif., 1986.
- [ELMASRI 89] RAMEZ ELMASRI, SHAMKANT B. NAVATHE, *Fundamentals of Database Systems*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1989.
- [HENDERSON 90] BRIAN HENDERSON-SELLERS, JULIAN M. EDWARDS, *The Object-Oriented Systems Life Cycle*. *Communications of ACM* 33:9, pp. 142-159, setembro de 1990.
- [HULL 87] RICHARD HULL, ROGER KING, *Semantic Database Modeling: Survey, Applications, and Research Issues*, *ACM Computing Surveys* 19:3, pp.201-260, Setembro 1987.
- [MATTOSO 90] ADRIANA L.Q. MATTOSO, *TABA OBJ: Um Ambiente de Desenvolvimento de Software com Orientação a Objetos*. Tese de M.Sc. em Eng. de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, Agosto de 1990.
- [MONTE 90] LUIZ C.M. MONTE, *O Modelo de Objetos do GEOTABA*. Exame de Qualificação para doutoramento em Eng. de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, 1990.
- [SHIPMAN 81] D. SHIPMAN, *The Functional Data Model and the Data Language Daplex*. *ACM Trans. Database Syst.* 6:1, pp. 140-173, março 1981.
- [TEOREY 86] TOBY J. TEOREY, DONGQING YANG, JAMES P. FRY, *A Logical Design Methodology for Relational Databases using the Extended Entity-Relationship Model*, *ACM Computing Surveys* 18:2, pp. 197-222, Junho 1986.