

**3º SIMPÓSIO  
BRASILEIRO  
de BANCO  
de DADOS**

23.25 de março de 1988  
Recife - Pernambuco

005.7406  
S612a  
1988

**ANAIS**

## UMA ANALISE DE BANCO DE DADOS ORIENTADOS PARA OBJETOS

Marina Teresa Pires Vieira  
Depto de Computação Univ. Fed. S. Carlos  
Rubens Nascimento Melo  
Depto de Informatica

PUC-RJ

## RESUMO

A tecnologia de Banco de Dados tão bem sucedida nas aplicações comerciais e administrativas recentemente tem se estendido para abranger aplicações menos convencionais como :Computer Aided Design, Automação de Escritorios, Banco de Dados Graficos, Engenharia de Software e outras. A "orientação para objetos" tem surgido como um paradigma integrador de varias vertentes de pesquisas da Informatica. Este trabalho apresenta uma analise sucinta de alguns exemplos de Sistemas de Gerência de Banco de Dados "Orientados para Objetos" em desenvolvimento em varios centros de pesquisa. O objetivo e' extrair a essência da "orientação para objeto" em busca de um Modelo de Banco de Dados mais expressivo e apropriado as aplicações nao convencionais.

## 1. INTRODUCAO

A tecnologia de Banco de Dados se desenvolveu principalmente na decada de 70 baseada no chamado "paradigma comercial". Os exemplos dos "Supridores e Pecas" e dos "Empregados e Departamentos" ficaram classicos e muitos dos topicos de pesquisa de Banco de Dados tinham como base tipos de aplicação semelhantes a desses exemplos basicos.

Essa tecnologia tão bem sucedida nas aplicações comerciais e administrativas recentemente tem se estendido para abranger aplicações menos convencionais como :Computer Aided Design, Automação de Escritorios, Banco de Dados Graficos, Engenharia de Software e outras [MEL087a]. O termo "orientação para objeto" tornou-se corriqueiro em varias areas de Informatica e , atualmente as pesquisas na area de Banco de dados tem se concentrado fortemente nesse topico. O proposito comum de tais pesquisas e' procurar uma forma de melhor capturar a semântica das novas aplicações da tecnologia de Banco de dados.

Precisa-se de Sistemas de Banco de dados que acomodem de forma natural essas aplicações orientadas para objetos. O modelo de dados dos sistemas deve conter um quadro de conceitos que possa expressar a semântica da aplicação. O modelo de dados determina a estrutura de dados , os operadores aplicaveis e as restrições de integridade a serem impostas. O que se deseja primariamente na "orientação para objetos" e' que o modelo de dados permita representar diretamente uma entidade do mundo real qualquer que seja sua complexidade por um unico objeto do Banco de dados incluindo a definição de operadores apropriados para explorar sua estrutura. Recentemente tem surgido na literatura da area de Banco de Dados algumas propostas de "Sistemas de Gerência de Banco de dados Orientados para Objetos". Neste trabalho serao

comentadas as principais delas procurando extrair os conceitos mais relevantes envolvidos nos modelos.

Na seção 2 é feita uma apresentação sucinta dos sistemas. Na parte 3 procura-se resumir os conceitos mais relevantes utilizados pelos diversos sistemas com o intuito de compor preliminarmente o quadro de conceitos apropriado para um Modelo de Banco de Dados "Orientado para Objetos".

## 2 SISTEMAS DE GERENCIA DE BANCO DE DADOS ORIENTADOS PARA OBJETOS

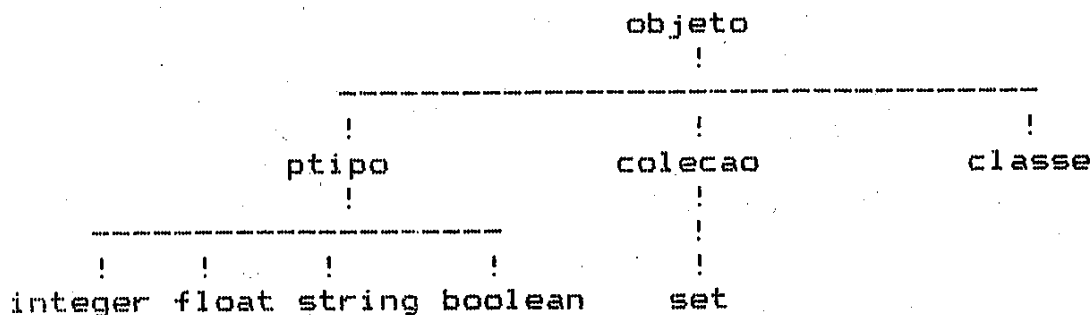
Varias instituições de pesquisa tem se empenhado nabusca de Sistemas de Gerência de Banco de dados "Orientados para Objetos". Varias abordagens podem ser percebidas nessas pesquisas. Por exemplo a influência das Linguagens de Programação Orientadas para Objetos, e em especial a da linguagem Smalltalk é certamente muito forte. Também a ideia de extensão de Sistemas de Gerência de Banco de Dados Relacionais tem importantes seguidores. Os exemplos de sistemas analisados brevemente a seguir, apresentam características dessas abordagens mencionadas acima.

### 2.1 SISTEMA IRIS

O modelo de dados do sistema IRIS [FISH87] baseia-se em tres construtores: *objetos*, *tipos* e *operacoes*. Os *objetos* representam entidades e conceitos do dominio da aplicação. Possuem existência propria no banco de dados e podem ser referenciados independentemente dos valores de seus atributos. Sao classificados por tipo: objetos compartilhando propriedades comuns pertencem ao mesmo tipo. Podem servir como argumentos e ou resultados de operações. Podem ser: *literais* (tais como cadeia de caracteres e numeros) que são diretamente representaveis ou *nao literais* que são representados internamente através de '*surrogates*'. Os *tipos* são coleções de objetos que compartilham propriedades comuns. Sao organizados em uma estrutura de tipo que suporta generalização e especialização. Assim, um tipo pode ser subtipo de um outro (supertipo), herdando as propriedades do super tipo. Um tipo pode possuir multiplos subtipos e multiplos supertipos. As operações (ou funções) são definidas sobre os tipos e são aplicaveis sobre as instancias dos tipos. E permitido, ao usuario, definir operações alem daquelas fornecidas pelo sistema.

### 2.2 ORION

No sistema ORION [BANE87] toda entidade conceitual é modelada como objeto cujo comportamento é encapsulado em metodos. Objetos semelhantes são agrupados em classes e se comunicam através de mensagens. Os objetos que pertencem a uma mesma classe são descritos pelas mesmas variaveis instanciadas e os mesmos metodos e respondem as mesmas mensagens. Em ORION, uma classe pode ter mais que uma superclasse. Assim, a hierarquia de classe é generalizada a um reticulado ('lattice'), isto é uma estrutura de grafo aciclico dirigido. ORION defini uma classe chamada OBJETO como sendo a raiz do reticulado. As outras classes são subclasses de OBJETO, como mostra a figura a seguir:



Um objeto composto é um objeto com uma hierarquia de objetos componentes exclusivos. A hierarquia de classe as quais os objetos pertencem é chamada hierarquia de objeto composto. Ela captura o relacionamento IS-PART-OF entre a classe pai e suas classes componentes.

### 2.3 ENCORE

O modelo de dados ENCORE [ZDON85],[HORN87] é um modelo semântico de alto nível. Neste modelo todos os objetos são instâncias de algum tipo que descreve o comportamento de sua instâncias. Portanto, um tipo envolve um conjunto de operações, um conjunto de propriedades e um conjunto de restrições que pertencem a qualquer uma das instâncias do tipo. Os tipos podem ser relacionados a outros através da propriedade especial IS-A que induz um relacionamento de herança entre tipos, podendo ainda haver herança múltipla (um tipo pode ter mais de um supertipo). Um tipo possui uma classe associada a qual os objetos do tipo pertencem automaticamente. Um tipo determina as operações que podem ser realizadas sobre as suas instâncias e especifica as propriedades que suas instâncias podem ter. O tipo mais geral é Entity, sendo que todos os outros tipos são direta ou indiretamente subtipos do tipo Entity. Uma classe contém todas as instâncias de um tipo particular, possuindo assim todos os objetos associados a este tipo. As classes são relacionadas entre si através da propriedade SUBSET-OF. Interessantemente, Tipos, operações e propriedades são todos objetos e como tais, possuem um tipo que descreve seu comportamento.

### 2.4 GEMSTONE

O sistema GemStone possui um subsistema de banco de dados que segue a linha do modelo de dados da linguagem Smalltalk. Seu modelo de dados envolve três conceitos principais: objeto, mensagem e classe. Um objeto tem uma interface pública e uma memória privada. Os objetos se comunicam através de mensagens, que são solicitações para que um objeto mude seu estado, retorne um valor ou realize alguma sequência de ações. As mensagens modelam o comportamento dos objetos. Um objeto responde a uma mensagem recebida, executando um método na linguagem OPAL. Os objetos que compartilham o mesmo formato e métodos são agrupados em classes e chamados de instâncias da classe. As classes são dispostas em uma hierarquia de classe onde cada subclasse herda o comportamento e estrutura de suas superclasses. GemStone combina as características de gerenciamento de bancos de dados tradicionais com um modelo de objetos semelhante ao do Smalltalk.

## 2.5 POSTGRES

O sistema POSTGRES [STON86] apresenta uma extensão do modelo relacional, incluindo novos tipos de dados (tipos abstratos de dados). Em particular permite procedimentos como um tipo de dado fundamental. É possível implementar novos tipos de dados que poderão participar em qualquer coluna de uma tabela do banco de dados. Podem ser adicionados a linguagem de consulta, procedimentos para avaliar operadores para os novos tipos criados. POSTGRES suporta dois tipos de dados que são procedimentos:

i) Tipo POSTQUEL. Pode-se declarar uma coluna de tabela como sendo do tipo POSTQUEL, o que implica que esta coluna contera uma sequencia de comandos que serao executados atraves do comando "execute".

ii) Procedimentos parametrizados. Todos os valores desta coluna usam o mesmo procedimento. Para que o procedimento nao seja armazenado em todas as tuplas da tabela ele pode ser armazenado separadamente.

## 2.6 SEMBASE

Um ambiente de aplicação do sistema SEMBASE [KING86] é constituído de uma coleção de objetos. Todos os objetos tem uma identificação unica ("surrogates") e são classificados de acordo como seu tipo e subtipo, estando organizados em hierarquias de tipo/subtipo. Os objetos podem ser: descritores, que são valores atômicos (inteiros, reais, booleanos e cadeia de caracteres) ou abstratos que são entidades não atômicas, definidas em termos de seus relacionamentos com outros objetos por meio de atributos. Um atributo é uma relação entre dois tipos. Pode-se definir um subtipo de um objeto especificando-se predicados sobre os valores de seus atributos, ou através de uma operação quando ele é dito arbitrário. Um subtipo herda todos os atributos do tipo pai e pode ter atributos adicionais. Algumas operações oferecidas pelo modelo são: Salvar e recuperar um Banco de dados, avaliar um atributo, e operações que se aplicam sobre objetos e atributos. (criar, retirar, adicionar...)

## 2.7 KNOS

KNOS (knowledge objects) [TSIC87] é um ambiente orientado para objetos que manipulam fragmentos de conhecimento com suas próprias regras apresentando a habilidade de objetos migrarem para novos ambientes e permitindo que os objetos adquiram novas operações dinamicamente. Um contexto possui um conjunto de KNOS que se comunicam através de um 'blackboard' lendo ou escrevendo mensagens nele. Um contexto está fisicamente associado a uma estação de trabalho. Assim, múltiplos contextos identificam várias estações de trabalho. Um KNO pode mover-se de um contexto para um outro. Cada contexto é controlado por um gerenciador de objeto que inspeciona o 'blackboard' local, aceita ou rejeita as solicitações de mudanças e decide se deve ou não se integrar com outros contextos (ou gerenciadores de objetos). Cada KNO é uma estância de uma ou mais classes. As classes especificam a estrutura inicial e o comportamento dos KNOs. A estrutura do KNO refere-se as variáveis instanciadas contidas no KNO. O comportamento do KNO é determinado pelas operações que ele pode realizar. Cada operação

e' composta de um conjunto de regras de produção em Zetalisp. As açöes do KNO são os átomos do comportamento do KNO. Elas são de cinco tipos:

1. Açöes locais ( Put,Get)
2. Açöes de comunicação (Import,Export)
3. Açöes existenciais (Spawn, Die, Move, Freeze, Unfreeze)
4. Açöes de aprendizagem (Act, Learn, Unlearn)
5. Açöes "limb" (Grow, kill, Ship, Teach, Unteach)

### 3. CONCEITOS PARA MODELAGEM DE BD ORIENTADO PARA OBJETOS

Um modelo orientado para objetos deve capturar de forma natural a semântica dos objetos envolvidos na aplicação. As aplicações ditas não convencionais envolvem objetos complexos cuja estrutura se configura como uma composição hierárquica de partes. Os componentes de um objeto complexo podem ser, por si, também objetos complexos, além disso pode haver uma composição recursiva de objetos e mesmo sobreposição de subobjetos da hierarquia. Essa estrutura complexa dos objetos, aliada a necessidade de representação do comportamento desses objetos requerem do modelo de dados novos conceitos além daqueles tradicionais. Os modelos de dados orientados para objetos sugeridos pelos sistemas apresentados acima parecem concordar em alguns conceitos básicos que devam estar presente em um "bom" Modelo de dados Orientado para Objetos". Esses conceitos são resumidos a seguir:

**OBJETO** Representa uma entidade do mundo real relevante para aplicação, possuindo as seguintes características: a) possui existência própria no Banco de dados sendo tratado como unidade de manipulação b) pode mudar de tipo ao longo de sua existência c) e' individualizado, isto é, os objetos são distintos entre si.

**TIPO** Esta' associado com um conjunto de operações que podem ser realizadas sobre instâncias do tipo, bem como com as propriedades que estas instâncias possuem. Os sistemas propostos permitem que sejam criados, pelos usuários, novos tipos de dados, além daqueles primitivos.

**CLASSES** Referem-se ao agrupamento lógico de objetos do mesmo tipo.. Permitem descrever num mesmo lugar (a classe), a estrutura e as operações que são desfrutadas por todas as instâncias da classe.

**OPERAÇÕES** São entidades dinâmicas que mudam o comportamento de um objeto. Cada objeto e' associado a um conjunto de operações que determina seu comportamento..

**FUNÇÕES** Alguns sistemas se utilizam de funções para expressar as propriedades de um objeto, os relacionamentos entre os objetos e operações sobre objetos.

**MENSAGENS** A comunicação entre os objetos podem ser feitas através de mensagens. Uma mensagem consiste de uma identidade do objeto receptor, o nome da mensagem particular e os argumentos para mensagem.

**ABSTRAÇÕES** Os modelos de dados orientados para objetos introduzem um conjunto semanticamente rico de primitivas de estruturação que suporta abstrações tais como:

Classificação, Generalização (ou Especialização) e Agregação

Classificação E' um mecanismo de abstração usado para agrupar entidades com propriedades semelhantes em classes.

Generalização/Especialização A generalização e' uma forma de abstração onde objetos semelhantes são abstraídos em um objeto generico de nivel mais alto. Os objetos constituintes são especializações do objeto generico.

Agregação E' um mecanismo de abstração que considera o relacionamento entre alguns objetos como um objeto de nivel mais alto. A agregação faz com que detalhes especificos dos objetos constituintes sejam ignorados quando se considera o objeto agregado.

As estruturas de tipo e de classe discutidas nos sistemas suportam generalização (e especialização). A hierarquia de classes permite reconhecer que um conjunto de classes esta associado a uma classe mais geral (superclasse).

A análise apresentada neste artigo e' preliminar. So'o aspecto "estrutural" dos dados como objetos foi abordado. Varios outros topicos (especialmente os de controle operacional tais como o controle de transações longas e gerência de versões) são muito importantes. Estudos sobre estes topicos e outros estão em andamento no departamento de Informatica da PUC-RJ visando a construção de um Sistema de Gerencia de Banco de Dados Orientado para Objetos, ferramenta central do Ambiente Integrado de Ferramentas para Sistemas Interativos (projeto EITIS) [MELO87b] atualmente em desenvolvimento no mesmo departamento.

## BIBLIOGRAFIA

- [BANE87] BANERJEE, J. et alli. -Data Model Issues for Object-Oriented Applications -ACM Transactions on Office Information Systems, vol.5 n.1 Jan 1987
- [FISH87] FISHMAN, D., H. et alli -Iris: An Object-Oriented Database Management System - ACM Transactions on Office Information Systems, vol.5 n.1 Jan 1987
- [HORN87] HORNICK, M.F.; ZDONIK, S.B. A Shared Segmented Memory System for an Object-Oriented Database - ACM Transactions on Office Information Systems, vol.5 n.1 Jan 1987
- [KING86] KING, R. A Database Management System Based on Object-Oriented Model -Proc. First International Conf. on Expert Database System. 1986
- [MAIE85] MAIER, D.; OTIS, A.; PURDY, A. Object-Oriented database Development at Servio Logic -IEEE Database Engineering, vol.88, n.4, 1985
- [MAIE86] MAIER, D.; STEIN, J.; OTIS, A.; PURDY, A. Development of an Object-Oriented DBMS -Proc ACM OOPSLA 1986
- [MELO87a] MELO, R.N. Banco de Dados Nao Convencionais VI JAI. VII Congresso da SBC 1987
- [MELO87b] MELO, R.N. EITIS :Um Ambiente Integrado de Ferramentas para Desenvolvimento de Sistemas Interativos. 1o Simposio de Engenharia de Software -Petropolis-Rj 1987
- [PURD87] PURDY, A.; SCHUCHARDT, B.; MAIER, D. Integrating an Object Server with Other Worlds - ACM Transactions on Office Information

Systems, vol.5 n.1 Jan 1987

[STON86] STONEBRAKER, M. Object Management in POSTGRES using Procedures -Proc. of International Workshop on Object-Oriented Database System, 1986

[TSIC87] TSICHRITZIS, D.; FIUME, E.; GIBBS, S.; NIERTRAZ, O. KNOS : Knowledge Acquisition, Dissemination and Manipulation Objects -ACM Transactions on Office Information Systems, vol.5 n.1 Jan 1987

[ZDON85] ZDONIK, S. B. Object Managements System for Design Environment -IEEE Database Engineering, vol.88, n.4, 1985