

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

23.25 de março de 1988  
Recife - Pernambuco

005.7406  
S612a  
1988

**ANAIS**

**SISTEMAS PARA GERENCIAMENTO DE BASES DE CONHECIMENTO :  
UMA EVOLUÇÃO DE SGBD**

**José Luis Braga**  
 Depto. de Matemática  
 UFV - Viçosa - MG  
 Membro do Projeto SAFO

**Rubens N. Melo**  
 Depto. de Informática  
 PUC - Rio de Janeiro - RJ

**RESUMO**

A transição que se observa, de processamento de dados para processamento de conhecimento, é acompanhada por um conjunto de idéias sobre como obter sistemas que processem conhecimento. Este artigo procura extrair a essência destas idéias, apresentando uma discussão de tópicos fundamentais sobre o assunto. O objetivo é visualizar uma arquitetura adequada para Sistemas de Gerência de Bases de Conhecimento (SGBC).

**1. INTRODUÇÃO**

Verifica-se hoje uma transição entre técnicas tradicionais de processamento de dados, normalmente usadas para solução de problemas bem definidos e organizados, com estrutura e semântica capturáveis e expressáveis através de modelos simples, baseados em conceitos tais como entidades, relacionamentos e atributos, para técnicas de processamento de conhecimento, que utilizam o conhecimento humano em determinados domínios como elemento básico de processamento.

Para que se entenda melhor o que significa processar conhecimento em um domínio, primeiro é preciso buscar uma caracterização do termo, ainda que de um ponto de vista puramente operacional, como a que se segue, que segmenta o conhecimento em duas categorias (SOWA, 1984) :

--conhecimento episódico, factual, ou extensional, correspondendo ao conhecimento sobre eventos e objetos do mundo real : "meu carro está com o freio estragado", "este texto foi editado em um microcomputador", etc...

--conhecimento heurístico ou intensivo (intensions) que corresponde a definições e regras de classificação de eventos, objetos e relações entre objetos, enunciando princípios gerais tais como "gatos são felinos", "homens são bipedes implumes e falantes", "se um veículo está com o freio estragado, leve-o imediatamente ao mecânico", etc...

O conhecimento heurístico, utilizado por especialistas de maneira instintiva ou mecânica é, de modo geral, superficial, pois é constituído por teorias de domínio, desenvolvidas ao longo da vida profissional do especialista, baseado em sua experiência. Já o neófito baseia seu conhecimento quase que exclusivamente em teorias profundas, sendo seus modelos de resolução de problemas mais operacionais, menos ricos em heurísticas.

Outras caracterizações estão disponíveis, como a proposta por (NEWELL,1982), que levanta uma Hipótese do Nível do Conhecimento:

**HIPOTESE DO NÍVEL DO CONHECIMENTO** - Existe um nível acima do nível simbólico, nos sistemas de computação, que é caracterizado por ter o conhecimento como meio de processamento, e o princípio da racionalidade como lei de comportamento.

O Princípio da Racionalidade, que é a lei de comportamento que rege sistemas que têm no conhecimento o elemento fundamental de processamento, representa uma relação entre conhecimento, objetivos a serem alcançados e as ações a serem executadas para este fim sem, contudo, especificar como deve ser feita a seleção de ações. É enunciada como :

**PRINCIPIO DA RACIONALIDADE** - Se um agente (elemento que usa o conhecimento como meio de processamento) tem o conhecimento de que uma de suas ações o levará a atingir o(s) seu(s) objetivo(s), então ele selecionará esta ação.

Nível, no caso, se refere a uma camada ("layer") de abstração, como é comum usar em computação de modo geral. E, na caracterização feita, o conhecimento constitui um nível autônomo, não estruturado e sem regras de comportamento e de composição bem definidas, ao contrário do que se espera quando um nível é definido.

Estas características tornam a Lógica (ou, de modo mais geral, as lógicas) uma ferramenta adequada para sua análise, uma vez que a lógica não impõe qualquer estrutura ao domínio, como é comum acontecer quando se usa outros métodos de representação, mais apropriados para implementações do que para análise. A lógica é um formalismo universal de representação (KOWALSKI,1985), situado abaixo do nível do conhecimento e implementável no nível simbólico, útil e poderoso na formalização e análise do conhecimento em um domínio.

Este uso da lógica, aliado à disponibilidade de um sistema de processamento de conhecimento baseado em lógica, como o Sistema SAPO (VIEIRA,1985), permite a construção de um modelo ou protótipo executável, levando a uma análise mais apurada, pois torna possível o teste de campo destes sistemas.

Um Sistema com Base de Conhecimento pode ser representado como na figura 1.

O **NÚCLEO** é a parte central do sistema, responsável pelo armazenamento, recuperação e manipulação do conhecimento, dando ao usuário externo a impressão de estar trabalhando com um sistema com um certo grau de competência em um domínio, que pode ser visto como um tipo primitivo de inteligência.

A **Base de Conhecimento** é um componente passivo, constituindo o depósito de conhecimento do sistema. Nela estão armazenados o conhecimento factual ou extensional e o conhecimento heurístico, ou especializado do domínio. Caracteriza-se por um alto grau de incompletude, pois não é possível capturar em um modelo todo o conhecimento sobre um domínio. O próprio conhecimento humano é incompleto, não sendo possível atribuir a um agente todo o conhecimento existente sobre um problema ou domínio particular (NEWELL,1982).

O **Mecanismo de Inferência** é um subsistema interno, geralmente acionado pela Interface de Consulta para resolver problemas propostos por usuários externos, ou por algum módulo interno. É este módulo que confere ao núcleo competência dedutiva, que depende do método ou métodos dedutivos utilizados. Esse mecanismo tem que ser capaz de manipular o conhecimento representado na base, seja qual for o domínio em questão, de uma forma chamada de **preservadora de valores verdade**, e aderente ao método de representação escolhido.

A **Interface de Consulta** é a parte do sistema que aparece para um usuário externo, que o utiliza como um auxiliar na tomada de decisões em sua área de atuação. É uma parte que deve ser projetada levando em conta tanto a estrutura do sistema de representação do conhecimento adotado, quanto técnicas modernas de projeto de Interfaces com o Usuário.

O **Sistema de Aquisição de Conhecimento** constitui a parte do sistema utilizada pelo chamado Engenheiro do Conhecimento (ou Analista do Conhecimento) para criar, depurar, aumentar, diminuir e testar consistência da base de conhecimento, tendo ainda uma parte de aquisição interna do conhecimento, oriundo de uma capacidade do sistema de fazer introspecção, aprendendo automaticamente com o uso.

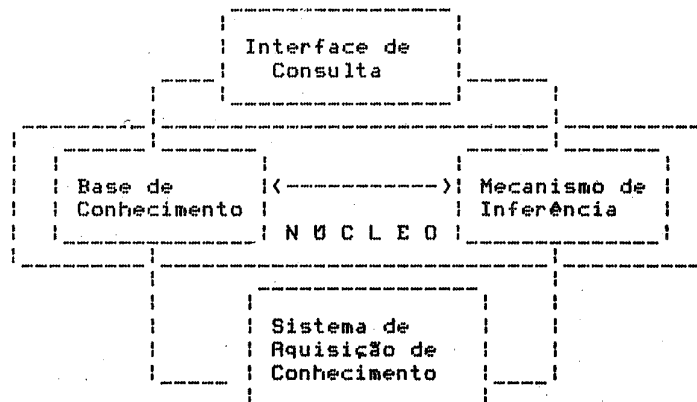


Fig.1 - Diagrama de Blocos Simplificado de um Sistema com Base de Conhecimento

## 2. SISTEMAS DE GERENCIAMENTO DE BASES DE CONHECIMENTO - SGBC

Um **SGBC - Sistema de Gerenciamento de Base de Conhecimento** - é um sistema de processamento que pode ser visto como uma evolução natural dos SGRD - **Sistemas de Gerenciamento de Bases de Dados**. Dentro deste contexto, que alia técnicas de Inteligência Artificial, próprias para processamento de semântica, com técnicas de gerenciamento de grandes volumes de

dados, herdadas das tecnologias desenvolvidas nas gerações anteriores, um SGBC deve ser capaz de **processar conhecimento**, oferecendo todo o serviço que se espera de um sistema desta natureza.

A figura 2 mostra, de forma esquemática, um resumo evolutivo das técnicas de processamento de dados. Linhas pontilhadas denotam desenvolvimento motivado por necessidade de capturar mais semântica da realidade, e linhas tracejadas representam o desenvolvimento motivado por questões ditas computacionais, ou seja, técnicas de implementação.

ESTAGIOS	PASSADO	PRESENTE	FUTURO
Requisitos de Inf. e Processamento	Proc.Comercial Simples	Proc. Dados Compartilhados, Estruturados, Rec. Erros, Segurança e Consistência	Sup. Decisão, Eng. e Proc. do Conhecimento
Modelos de Dados	Modelos de Dados Clássicos	Modelos de Dados Semânticos	?????
SGBD Eficiente	Segunda Geração de SGBD	Terceira Geração de SGBD	Sistema de Gerenciamento de Base de Conhecimento (SGBC)

Fig.2 Desenvolvimento Computacional e Semântico em Bancos de Dados (MYLOPOULOS,1985)

O corpo de conhecimento sobre SGBD, classificado na figura 2 como "SGBD de segunda geração", denominado de **teoria computacional** (MYLOPOULOS,1985), é formado, principalmente, por pesquisas em Teoria de Bancos de Dados, fortemente ligado ao formalismo introduzido no Modelo Relacional, Tópicos sobre Implementação e Projeto de Sistemas de Bancos de Dados, e Modelos de Dados e Linguagens.

O desenvolvimento de modelos de dados da segunda geração de SGBD foi mais no sentido de obtenção de técnicas eficientes de implementação dos modelos de dados existentes, que eram vistos como modelos de implementação de bancos de dados.

A terceira geração de SGBD trouxe uma inversão nesta tendência: o SGBD passou a ser encarado como meio, tendo o foco

de atenções se desviado de detalhes de implementação para técnicas de análise mais apurada da realidade, na tentativa de capturar mais semântica (contexto), através do uso de modelos de maior poder expressivo, denominados **modelos semânticos**. Exemplos que podem ser citados são o de **Entidades e Relacionamentos** (CHEN,1976), ou suas extensões como o **ERE - Entidades e Relacionamentos Estendido** (CERI,1983). Existem implementações, como o sistema PROJET (NARDELLI Jr.,1987), que permitem construir o modelo semântico com apoio de "software", obtendo um protótipo que, após refinado e testado, pode ser automaticamente implementado em um sistema de Banco de Dados Relacional.

Existe uma gradação suave entre sistemas de bancos de dados puros, vistos como casos particulares e restritos de bases de conhecimento, em que mecanismos de busca puros são suficientes para implementar uma espécie restrita de consequência lógica, e sistemas completos de base de conhecimento, com componente de dados e componente de conhecimento, exigindo tanto uma parte de busca em dados, quanto um mecanismo sofisticado de inferência, para obtenção de consequências lógicas (LEVESQUE,1986), representada pela figura 3:

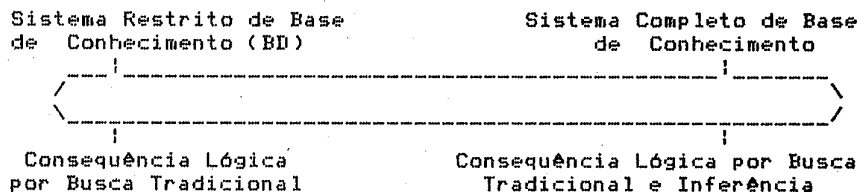


Fig.3 Espectro de Classificação de SGBC

As propostas disponíveis na literatura sobre como obter um SGBC podem ser resumidas como na figura 4. Todas elas giram em torno do problema de como obter sistemas que aliem um mecanismo dedutivo com capacidade de armazenamento e recuperação de grandes volumes de dados.

Um SGBC Homogêneo Elementar nada mais é que um Sistema de Programação usando Lógica, como o PROLOG. Como exemplo de uma base de conhecimento usando o micro-PROLOG (BRAGA,1987), o banco de dados "canônico" de fornecedores, peças e fornecimentos (atributos têm o significado usual) ficaria:

ESQUEMA CONCEITUAL

```
fornecedor(número, nome, classe, cidade)
peças(código, nome, cor, peso)
fornecimento(numfornecedor, códpeça, quantidade)
```

A base extensional a seguir servirá como exemplo:

```
fornecedor (f1 silva 20 rio)
fornecedor (f2 soares 10 salvador)
```

```

fornecedor (f3 menezes 30 manaus)
fornecedor (f6 vieira 20 patos)

fornecimento(f2 p1 100)
fornecimento(f3 p2 30)

peças(p1 parafuso cinza 1)
peças(p2 prego cinza 1)

```

Consultas convencionais podem ser dirigidas ao sistema, tais como (respostas seguem-se a cada consulta, quando for o caso):

.. quais os números e nomes dos fornecedores de manaus ?

```

which (x y z : fornecedor (x y z manaus))
f3 menezes

```

.. qual o número e nome dos fornecedores que estão fornecendo alguma peça ?

```

which( x y z : fornecedor (x y z Z1) and
fornecimento ( x y1 y2))
f2 soares
f3 menezes

```

Ainda como exemplo, seja a consulta (LLOYD,1983) fornecer os nomes dos fornecedores que fornecem a peça p2, expressa em SQL como

```

select fornecedor.nome
from fornecedor, fornecimentos
where fornecedor.número =
fornecimento.numfornecedor
and fornecimento.codpeça = "p2"

```

que pode ser submetida ao PROLOG com a consulta

```

which (y z : fornecimento(x p2 y) and
fornecedor (x y z z1))

```

Uma crítica a esta visão de implementação de SGBC é que o PROLOG só faz o gerenciamento de dados residentes na memória principal. Não tem recursos de espécie alguma para manipular dados na memória secundária, e muito menos de organização sofisticada que permita o tipo de acesso requerido por um banco de conhecimento, em tempo de resolução de uma consulta.

SGBC's Homogêneos Avançados dizem respeito a sistemas PROLOG com recursos de armazenamento e recuperação de informação localizados em memória secundária.

De modo geral, a parte de regras de uma base de conhecimento pode residir na memória principal durante o tempo do processamento, pois uma de suas características marcantes é o pouco volume de código que a constitui. Já a parte de proposições

sobre fatos, que ocorrem em grande volume, necessitam residir permanentemente em memória secundária, sendo carregadas em memória principal apenas quando necessário.

Uma das idéias correntes é expandir o PROLOG adicionando a ele os recursos necessários, tais como métodos de acesso próprios, e os disponíveis nos sistemas de gerenciamento de arquivos dos SGBD. Esta visão da solução do problema tem uma vantagem: as facilidades adicionadas ao PROLOG passam a ficar disponíveis a todas as aplicações que o utilizarem.

**PROPOSTAS EVOLUTIVAS** - Fazem a transição de maneira suave, aproveitando sistemas já disponíveis.

**HOMOGENEAS** - Usam sistemas já existentes, expandindo-os com novas técnicas.

**ELEMENTAR** - Sistema de Programação Lógica com Dados em Memória Principal (PROLOG, p.ex.)

**AVANÇADO** - Sistema de Programação Lógica combinado com Método de Acesso a Memória Auxiliar

**HETEROGENEAS** - Usam sistemas já existentes, combinando-os por meio de interfaces adequadas.

**ACOPLAMENTO FRACO** - Compilação de Consultas aliada a um SGBD convencional.

**ACOPLAMENTO FORTE** - Interpretação de Consultas

**PROPOSTAS REVOLUCIONARIAS** - Criam métodos e idéias novos, nada aproveitando dos sistemas já existentes.

**Fig. 4** Propostas para Obtenção de SGBD (MISSIKOFF, 1984)

Uma outra solução é, em um sistema de base de conhecimento particular, adicionar os recursos de gerenciamento de arquivos implementados especialmente para aquela aplicação. Isto leva a implementações eficientes e dedicadas.

Estas soluções tendem a produzir sistemas muito grandes, pois grande parte das rotinas de gerenciamento de arquivos de um SGBD acabam sendo implementadas, aumentando muito a quantidade de código.

Dadas as dificuldades de implementação e limitações das soluções homogêneas, partiu-se para a tentativa de novas soluções usando "software" disponível no mercado e a idéia de fazer



acoplamento entre pacotes diferentes. Os requisitos são bem definidos: é necessário um sistema dedutivo, como o PROLOG, e um sistema de gerenciamento de volumes grandes de dados, como um SGBD de segunda ou terceira gerações.

Dois problemas são abordados com este enfoque (LLOYD,1983):

- indexação de cláusulas em memória secundária, ou seja, qual a melhor maneira de recuperar cláusulas, que são a forma de representação usada pelas implementações do PROLOG, da memória secundária?

- otimização de consultas, ou seja, qual a melhor sequência de operações a serem executadas para obter resposta a um dado problema?

O problema da indexação de cláusulas não é de solução trivial e eficiente, embora haja soluções conhecidas, herdadas de pesquisa em SGBD. A questão é que, dada uma relação qualquer, não há determinismo de argumentos de consulta, significando que os argumentos podem ser tanto de entrada quanto de saída. Isto significa que, em uma consulta, pode ser necessário recuperar cláusulas referentes a ocorrências de relações, usando qualquer um dos argumentos como elemento de busca.

Uma solução, herdada de bancos de dados, é usar os chamados arquivos de inversão, invertendo completamente, em todos os campos e nome de relação, todas as relações de uma base. Mas, esta solução resolve um problema e cria outro mais complicado, pois carrega consigo todos os problemas herdados de seu uso em SGBD.

Outra solução é o uso de uma técnica de acesso por espalhamento com múltiplas chaves, em que cada um dos argumentos (chaves secundárias) de uma relação tem sua própria função de espalhamento associada (hashing function). Em um acesso, os valores das funções de espalhamento para cada um dos argumentos apresentados na consulta são calculados e combinados de maneira tal que o valor final obtido possa ser usado como índice de acesso a toda a cláusula (LLOYD,1983).

O problema da otimização de consultas se refere à questão de escolher a melhor sequência de execução, para obter uma resposta. Melhor, no caso, se refere à sequência que minimize o número de acessos à memória secundária, trazendo para a memória principal apenas o necessário a cada passo de execução, e tirando o máximo de proveito do conhecimento já disponível na memória principal.

Quando as consultas são mais complexas, representadas por fórmulas envolvendo conectivos, a escolha da melhor sequência não é trivial. Exige um sistema de análise de consulta sofisticado, ele próprio baseado em conhecimento tanto sobre o domínio, quanto sobre o método de acesso empregado, dentre outros fatores.

Uma idéia é usar Sistema Com Acoplamento Fraco, que tende a manter as características de cada um dos dois sistemas acoplados: o sistema dedutivo só faz deduções lógicas, e o SGBD gerencia os volumes de dados em memória secundária (MISSIKOFF,1984), sendo indicada nos documentos sobre o Projeto

#### Japones de Quinta Geração.

Ela é chamada de **solução compilada**, pois podem ser distinguidas duas fases distintas:

- na primeira fase, a consulta dirigida ao sistema é resolvida pelo sistema dedutivo, que gera uma nova expressão (expressão compilada), representando a consulta original, usando operações próprias do SGBD. Nesta fase pode ocorrer a otimização da consulta, pois o sistema compilador pode usar um subsistema baseado em conhecimento para isto;

- na segunda fase, a expressão gerada na primeira fase, contendo apenas operações próprias do SGBD e se referindo a fatos, é executada pelo SGBD, que fornece a resposta final.

Uma vantagem desta solução é a de que SGBD disponíveis no mercado podem ser usados, não tendo que ser adaptados. A única providência necessária é a criação de uma interface (protocolo) de comunicação adequado, e uma desvantagem é que o Sistema Dedutivo enxerga o banco de dados através de uma "janela" (MISSIKOFF, 1984), que funciona como uma visão congelada dos dados armazenados. Se o banco de dados é sujeito a alterações constantes, como em um sistema distribuído, esta visão congelada (snapshot) pode ficar desatualizada, ocorrendo problemas de consistência.

Uma outra idéia é usar **Sistemas Com Acoplamento Forte**. Neste enfoque, também chamado de **solução interpretada**, a interação entre o Sistema Dedutivo e o SGBD ocorre em qualquer instante durante o processo de dedução lógica. A parte factual da base de conhecimento é vista como uma extensão natural da parte de regras, sendo as duas usadas pelo sistema de forma indistinta.

Neste caso, a responsabilidade por otimização fica por conta do mecanismo de inferência, que passa a reinar absoluto como gerador de decisões internas. Para tanto, ele precisa ser um mecanismo sofisticado, capaz de escolher, durante o processo de dedução, qual a relação a ser usada, baseado em conhecimento sobre o domínio em questão e também em conhecimento sobre o SGBD que está sendo usado, de detetar laços de execução, de saber que uma determinada tupla já está disponível para uso na memória, sendo desnecessário um novo acionamento do SGBD, etc...

O problema de inconsistência que pode surgir na solução com acoplamento fraco fica eliminado, pois a interação se dá em tempo de execução. Para que todo o conjunto fique eficiente, é necessário usar um SGBD de grande eficiência no acoplamento.

Classificam-se como **propostas revolucionárias** aquelas que significam uma visão completamente nova, não tentando aproveitar nenhum dos sistemas já prontos e disponíveis no mercado. Representam tópicos de pesquisa na área, sendo sistemas sofisticados de processamento, que se utilizam da tecnologia disponível tanto em SGBD quanto em IA para atingir seus objetivos.

Uma idéia, proposta por (MANOLA, 1986) é a de usar a filosofia de **Sistemas Abertos (open systems)** (HEWITT, 1985). Estes são sistemas de processamento distribuído de conhecimento que levam em consideração a evolução e mudança contínuas no tempo, que lhes permite adaptar-se a novas formas de processamento,

equipamentos e requisitos externos, evoluindo para as novas necessidades. Permitem a tomada de decisão descentralizada, que impede que um elemento da rede interfira nas decisões locais de outro elemento ou componente, reconhecendo uma inconsistência permanente entre bases de conhecimento de cada nó do sistema, causada por problemas de privacidade e controle localizado de aquisição de conhecimento.

A descentralização das bases de conhecimento, organizadas como federação dificulta enormemente o problema de manutenção de consistência (propagação de alterações). Além disto, a inadequabilidade da hipótese do mundo fechado torna-se presente, pois o universo de conhecimento, agora, é muito amplo, aproximando-se mais da realidade.

Decisões são tomadas e aplicadas por concordância dos elementos da rede, sendo as soluções de problemas buscadas mais por um tipo de exploração do conhecimento distribuído do que por uma busca exaustiva no espaço de soluções possíveis, como é feito nos sistemas de programação em lógica.

### 3. CONCLUSOES

Não há consenso sobre qual a arquitetura ideal para um SGBC. Existem idéias isoladas, a maior parte delas tendendo a obter de maneira rápida uma implementação que permita conduzir experimentos e, de uma maneira "de baixo para cima", chegar a um modelo geral.

A maior dificuldade provém do fato de que não se consegue sequer caracterizar, de modo claro, o que é conhecimento, e muito menos definir que tipo de conhecimento deve ser embutido em um SGBC, e como esse conhecimento deve ser capturado e processado. Métodos tradicionais de análise de dados não se aplicam a conhecimento, devido à ausência de estrutura. Outros problemas surgem, tais como o de adequar mecanismos de representação e inferência a cada tipo de domínio, baseado exclusivamente em suas características estruturais. Além disto, a implementação final de Sistemas de Base de Conhecimento ainda é um tópico não explorado, estando os sistemas conhecidos ainda na fase de prototipação.

O sistema ideal será aquele que proporcionar ao usuário um ambiente completo de desenvolvimento, fornecendo recursos para que seja feita análise e classificação automática de domínios de conhecimento, seleção de método de representação de conhecimento adequado, prototipação, teste e geração do sistema final.

### 4. BIBLIOGRAFIA

BRAGA, J.L. Uma avaliação crítica do PROLOG como linguagem para processamento de conhecimento. Rio, PUC, Fev.1987. (exame de qualificação).

BRODIE, M.L., MYLOPOULOS, J. On Knowledge Based Management Systems. New York, Springer-Verlag, 1986.

- CARVALHO, R.L. e outros. Processamento do Conhecimento. Tandil, II EBAI, Fev. 1987.
- CERI, S.(ed.) Methodology and Tools for Data Base Design. North-Holland, 1983.
- CHEN, P.P-S. The entity-relationship model : toward a unified view of data. ACM TODS 1(1):9-36, March 1976.
- HAYES, P. In defence of logic. Proc. IJCAI-5, p.559-565, 1977.
- HEWITT, C. The challenge of open systems. RYIE 10(4):223-242, April 1985.
- HOPSTADTER, D.R. Godel, Escher, Bach : An Eternal Golden Braid. Vintage Books, 1980.
- KERSCHBERG, L.(ed.) Proceedings of the First International Workshop on Expert Database Systems. Kiawah Island, South Carolina, Oct. 24-27, 1984.
- KOWALSKI, R.A. The limitations of logic. Proc. Workshop on KBMS, Chania, Crete, June 1985.
- LLOYD, J.W. / An introduction to deductive database systems. The Australian Computer Journal, 15(2):52-57, May 1983.
- MANOLA, F., BRODIE, M.L. On Knowledge based system architecture. In: (BRODIE, 1986).
- MISSIKOFF, M., WIEDERHOLD, G. Towards a unified approach for expert and database systems. In : (KERSCHBERG, 1984).
- MYLOPOULOS, J. On knowledge based management systems. In : (BRODIE, 1986).
- NARDELLI Jr., D. CAI para projeto de sistemas de informação. Rio, IME, 1987. (Tese de Mestrado).
- NEWELL, A. The knowledge level. AI 18, p.87-127, 1982.
- SOWA, J.P. Conceptual Structures : Information Processing in Minds and Machines. Addison-wesley, 1984.
- VIEIRA, N.J. SAFO : Manual do Usuário. Rio, PUC, 1985.