

Armazenamento de Informações - SBU  
Sistemas: Informações  
Modelagem conceitual  
Bancos: Dados inteligentes  
ENPq 1.03.03.00-6

## Sistema de Informação Total : Gerência de Dados e Conhecimento

José Mauro Volkmer de Castilho

Mara Abel\*

### RESUMO

Sistemas de Informação, num sentido amplo, são todos os sistemas que permitem o manuseio e consulta seletivos da dados, conhecimentos, textos e imagens, para apoio à construção de planos, pareceres e diagnósticos, em alguma área de aplicação. As atuais tecnologias de projeto e desenvolvimento de sistemas de informação não são capazes de tratar o problema de modelar e implementar um sistema único que gerencie todos estes tipos de informação. Neste trabalho se discute o projeto e o desenvolvimento de SIT's - Sistemas de Informação Total, propondo um modelo conceitual de informações adequado, e propondo uma forma de integrar vários sistemas em computador sob uma mesma camada de "software", constituindo o Sistema Total implementado.

### ABSTRACT

Information systems, in a wide sense, are all systems that allow selective handling and retrieval of data, knowledge, text pieces and images, to support the preparation of plans, reports and diagnosis, in several application areas. Current design and development technologies for computer-based information systems are not able to deal with the problem of modeling and implementing a system that could manage all the above mentioned kinds of information. This work discusses the problem of design and development of SIT's - Total Information Systems, and proposes a convenient conceptual information model, together with a method to integrate component specialized information systems, under a single "software" interface.

**PALAVRAS-CHAVE:** sistemas de informação, modelagem conceitual, integração de sistemas de informação, bancos de dados inteligentes.

## 1 INTRODUÇÃO

Considerando uma definição ampla, pode-se dizer que pertencem à classe dos *sistemas de informação* em computador, todos os sistemas de "software" que, em operação e depois de estimulados por uma *consulta*, comando ou operação de um *utilizador*, apresentam, por algum meio de comunicação máquina-homem (som, imagem ou texto em vídeo, impressão em formulário ou folha, etc), uma *resposta*, a representação externa de um conjunto de dados que corresponda à consulta feita e que vai aumentar o grau de conhecimento do utilizador sobre determinado assunto, permitindo que ele possa desempenhar melhor alguma *tarefa* ou serviço.

Com esta definição, são sistemas de informação: os Sistemas de Bancos de Dados, tais como definidos em [3]; os chamados *Sistemas de Recuperação de Informações (IRS)* [10], usados para recuperação de informações bibliográficas; os Sistemas Especialistas [5], nas suas aplicações como sistemas de classificação, diagnóstico ou projeto.

A maioria das profissões do setor econômico de Serviços, e muitas das profissões de nível superior do setor Industrial, para serem bem exercidas dependem fortemente de acesso fácil, rápido, e inteligente, a informações complexas, de grande volume, que podem ser registros, texto livre, imagens, mapas, gráficos, ou sons. Exemplos de áreas de atividade onde a disponibilidade de informação é essencial: planejamento urbano e rural; avaliação de impacto ambiental de lançamento de efluentes sólidos ou gasosos; mercado de capitais, investimentos; jornalismo; elaboração de relatórios de prospecção geológica; elaboração de diagnósticos médicos. Em geral, atividades em que alguém (um perito ou especialista), precisa trabalhar com um objetivo em mente, e com um conjunto de informações volumoso e heterogêneo, para produzir um plano, proposta, diagnóstico ou parecer conclusivo.

Isoladamente, é possível gerenciar cada um dos diferentes tipos de informação, utilizando sistemas de bancos de dados, sistemas gráficos, de hipertextos, hipermeios ou sistemas especialistas. Neste caso, a tarefa do utilizador não é fácil. É necessário conhecer e utilizar linguagens diversas, baseadas em paradigmas de representação de informação diferentes uns dos outros, e lançar mão de expedientes pouco técnicos, tais como anotar em folhas de papel dados obtidos de um sistema para usar como subsídio em consultas a outros sistemas. Em suma, um ambiente de trabalho complicado, não confiável, e que exige muito esforço do seu utilizador.

O objetivo deste trabalho é explorar os requisitos e soluções para implementação de um sistema que integre as capacidades desses diversos sistemas em uma única plataforma, sob um único modelo conceitual. Este modelo deve ser poderoso o suficiente para representar conhecimento sob suas diversas formas, e padronizado o suficiente para não afetar a performance no acesso e recuperação de um grande volume de informações.

Para identificar os requisitos necessários para o projeto e construção de um tal "sistema de informação total" (SIT), como poderia ser chamado este sistema de informação integrado, instalado numa plataforma de "hardware" adequada, veja-se, por exemplo, as atividades e o conjunto de recursos de informação que deveriam estar à disposição de um *Agente de Viagens* perito. Este Agente tem como função sugerir a pessoas conhecidas ou



cadastradas, pacotes ou roteiros de viagens que elas gostariam de fazer, ou atender pessoas que o procurem para planejar e tomar as providências necessárias para a realização de uma viagem. Este exemplo é de fácil compreensão e dá uma boa idéia das características principais e do grau de complexidade dos sistemas de informação cujo estudo é o objeto deste trabalho.

O texto a seguir está dividido em mais duas seções, além da introdução e das conclusões. A Seção 2 apresenta os requisitos do sistema a partir das necessidades das informações que irá gerenciar; a seção 3 discute a modelagem conceitual e a implementação de um SIT.

## 2 REQUISITOS DO SISTEMA E CARACTERÍSTICAS DAS INFORMAÇÕES

As funções de um Agente de Viagens, para fins de identificação de registros do sistemas são:

(a) gerenciar um volume de informações que incluem dados sobre cidades e locais turísticos, informações sobre empresas de transporte, hotéis, órgãos de financiamento, além do cadastro de clientes;

(b) identificar em cada cliente em potencial um determinado perfil, ou combinação de perfis, para sugerir o *pacote*, que é como se denomina uma viagem com seu roteiro e seu conjunto de atividades e programas;

(c) *construir* pacotes, para grupos ou pessoas, a partir de um conjunto de informações sobre lugares, eventos, e outros recursos, para satisfazer demandas de clientes, atuais ou previstas.

O Sistema de Informação Total tem como objetivo tornar disponível todo o conjunto de informações associado às funções do Agente listadas acima, bem como auxiliar no processo de decisão. A construção de uma tal *estação de trabalho* do Agente não apresenta atualmente problemas do ponto de vista de "hardware": é possível montar, a preços razoáveis, plataformas constituídas por estações de trabalho básicas, com suficiente quantidade de memória local, e periféricos como unidades de disco rígido de grande capacidade (para os bancos de dados) ou unidades de discos óticos (para os arquivos de imagens e arquivos bibliográficos de grande volume).

O desenvolvimento do sistema com as características de inteligência e eficiência esperadas não é uma tarefa trivial e os maiores problemas parecem estar em dois pontos: (1) como modelar e projetar a base de informações de tal sistema? (2) como integrar sob uma mesma "casca" a inteligência e flexibilidade, comum às ferramentas de processamento simbólico, e a robustez e eficiência dos sistemas de gerência de bancos de dados.

A dificuldade de modelagem das informações para um SIT está principalmente baseada nas características não triviais (em termos de processamento eletrônico de dados) das representações destas informações. Ou seja, por sua natureza, as aplicações às quais se aplicam os SIT necessitam maior riqueza de detalhes na modelagem das informações e um conjunto de possíveis consultas mais sofisticado, sem abrir mão de eficiência no

gerenciamento de um grande volume de dados. Um resumo das características das informações é mostrado abaixo:

(1) Complexidade das informações: é preciso representar e armazenar, sem perda de conteúdo, informações que estão associadas entre si por relacionamentos complexos, como: prioridade (ordem de geração das informações); dependência de existência entre informações; herança de propriedades comuns a membros de um conjunto. A modelagem destas relações exige modelos de dados que sejam semanticamente ricos.

(2) Diferentes tipos e formas de representação de informações: o sistema deve gerenciar representações tão diferentes como tabelas e tuplas, conhecimento expresso em regras ou "frames", textos de tamanho variável, imagens, sons, gráficos, mapas, plantas, fotos.

(3) Volume de informações: neste aspecto, os SIT se assemelham a sistemas de bancos de dados de muito grande porte, com a diferença de que é preciso ter mecanismos adaptados para manipulação e acesso eficiente aos dados não tabulares, bem como mecanismos para proteção e recuperação em caso de falha para dados destes outros tipos.

(4) Forma de acesso e manipulação: como consequência dos aspectos anteriores, há necessidade de prover mecanismos eficientes e inteligentes, que reconheçam os diferentes tipos de informação e selecionem os métodos adequados, para acessar as informações de maneira homogênea e confortável ao utilizador.

### 3 ARQUITETURA DO SISTEMA

A caracterização de um Sistema de Informação Total baseia-se em três premissas:

(1) A captação da realidade e sua representação da forma mais completa possível, obtidas através da utilização de um modelo conceitual semanticamente rico;

(2) a inteligência dos mecanismos de consulta e gerência da informação, obtida através da utilização de uma ferramenta de processamento simbólico para gerar uma representação interna do modelo conceitual e implementar a forma de raciocínio;

(3) a eficiência e segurança no armazenamento e recuperação de informações através da utilização de um sistema de gerência de banco de dados, e sistemas de gerência de textos e imagens para armazenar as instâncias das informações.

O modelo conceitual para um SIT é definido formalmente através de uma linguagem de representação das informações e utilizado como uma ferramenta para análise dos dados. A versão "física" do modelo é feita utilizando uma linguagem simbólica, na qual se busca implementar os objetos modelados, suas propriedades e relações sem perda de conteúdo. As instâncias das informações são armazenadas em um banco de dados relacional ou em um sistema de gerência de imagens e textos. Os dois componentes-chave do sistema (sistema simbólico e sistema de banco de dados) são integrados em um sistema único através de um esquema de mapeamento que associa a forma de representação de dados de um componente para o outro.

Os componentes do sistema são representados esquematicamente na figura 1.



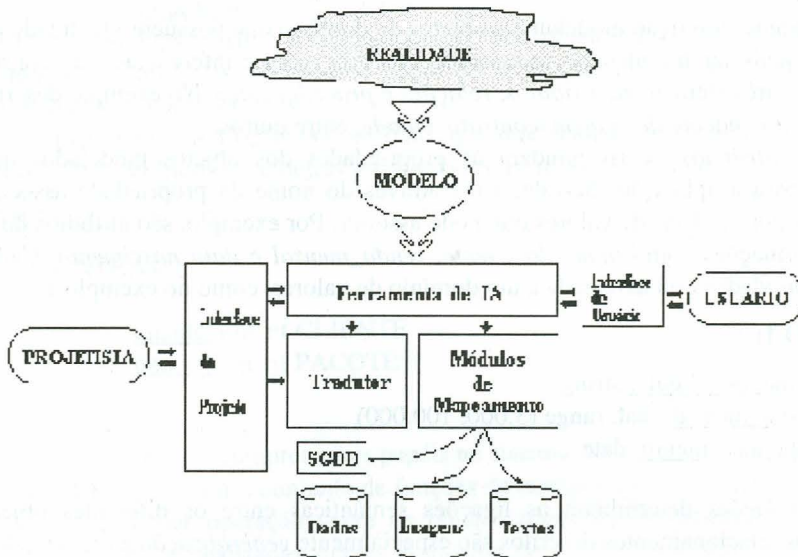


Figura 1 - Arquitetura de um Sistema de Informação Total.

### 3.1 O MODELO CONCEITUAL

O SIT é um sistema cuja base de informações é na realidade a agregação de várias bases: de dados, de conhecimentos, de textos, de imagens. Para cada uma destas bases existem metodologias próprias de coleta ou aquisição e modelagem conceitual de informações. Mas, na fase de modelagem conceitual da base de informações do SIT, ainda não se tem a definição de quais conjuntos de informações vão se encaixar em qual tipo. Então, é preciso que o modelo conceitual de informações, ou seja, o conjunto de noções segundo as quais se vai organizar as descrições das informações da base, a nível conceitual, seja suficientemente geral e poderoso para permitir a descrição das características de todos os tipos.

O modelo proposto neste trabalho é uma adaptação do modelo de frames, baseado na modelagem de objetos, seus atributos e os procedimentos ligados a eles. É semelhante, em alguns aspectos ao FORM, proposto em [6], que é um modelo de dados que combina características dos modelos de "frames" [9] e de modelos orientados a objetos [4].

O modelo é menos flexível do que o modelo de frames utilizados em muitas aplicações de Inteligência Artificial [8][7]. As restrições impostas ao modelo buscam garantir a correspondência eficiente com o modelo relacional.

Na abordagem proposta, são dois os elementos básicos de modelagem: os *frames-descrição* e *frames-regra*, que guardam os conhecimentos do tipo *objetos* e do tipo *habilidades* segundo descrito em [1].

### 3.1.1 FRAME-DESCRIÇÃO

Os frame-descrição modelam os objetos do domínio que possuem identidade própria e sobre os quais serão realizadas as consultas, modificações e inferências. São construídos utilizando-se três elementos: *atributos*, *relações* e *procedimentos*. No exemplo descrito, são objetos: *cliente*, *pacote de viagem*, *contrato*, *cidade*, entre outros.

Os *atributos* correspondem às propriedades dos objetos modelados que são relevantes para a aplicação. São descritos através do nome da propriedade associada ao objeto e o valor ou listas de valores que pode assumir. Por exemplo, são atributos do objeto *cliente* informações como *nome do cliente*, *renda\_mensal* e *data nascimento*. Cada uma dessas propriedades está associada a um domínio de valores, como no exemplo 1:

#### EXEMPLO 1:

nome do cliente: string  
renda mensal: real, range (3.000, 100.000)  
data nascimento: date

As *relações* determinam as ligações semânticas entre os diferentes objetos do domínio. Os relacionamentos descritos são especialmente *generalização* e *associação*.

Uma *generalização* é descrita através do atributo *é\_um* associado ao objeto especializado, e cujo valor é o nome do objeto que corresponde ao frame mais geral. São associações de cardinalidade um-para-muitos. Por exemplo, para a aplicação descrita define-se o objeto EVENTO que será a generalização de cada uma das atividades incluídas no pacote do cliente. Assim, uma ida a um museu, ou uma feira, ou um passeio de barco, correspondem a diferentes modalidades (ou *especializações*) deste objeto. Como especializações, estes objetos podem ser descritos por, no mínimo, o mesmo conjunto de propriedades que descreve o frame mais geral, mas essas propriedades assumem valores diferentes para caracterizar os objetos (ou seja, as especializações *herdam* as propriedades do frame mais geral). No exemplo 2, as especializações de EVENTO também são descritas pelas propriedades *nome* e *local*, porém VISITA MUSEU inclui o atributo *tipo*, e PASSEIO inclui *duração média*.

#### EXEMPLO 2:

EVENTO

nome: string  
local: string

VISITA MUSEU

é um: EVENTO  
tipo: one of {"arte", "histórico", "ciência"}

PASSEIO

é um: EVENTO

duração-média : integer

Uma *associação* define relacionamentos entre objetos distintos da aplicação. São declarados como atributos, cujo nome identifica o tipo de relação, e cujos valores são nomes de outros frames-descrição.

O objeto CONTRATO possui atributos que o relacionam a um determinado cliente e a um pacote específico, entre outras propriedades listadas no exemplo 3.

### EXEMPLO 3:

#### CONTRATO

...  
cliente: one of CLIENTE  
pacote: one of PACOTE  
 ...

Os *procedimentos* cumprem dois papéis no sistema:

- O de constituir o conjunto de funções de manipulação dos objetos, ou seja, toda e qualquer operação sobre os objetos (consulta, modificação, exclusão) só podem ser realizadas através destas funções. Esses procedimentos garantem o *encapsulamento* dos objetos em relação às suas funções de manipulação, auxiliando a integridade e independência dos dados e estão associados ao próprio objeto a ser manipulado.
- O de calcular, obter ou verificar os valores a serem dados aos atributos das instâncias do banco de informações. Esses procedimentos podem ser simples funções de cálculo como no exemplo 4 com o atributo *idade*, ou programas complexos que realizam interação com o utilizador para deduzir a informação desejada. São associados, quando necessário, à declaração dos atributos dos objetos.

No exemplo 4, os procedimentos *incluir\_cliente*, *mostrar\_cliente* descrevem os métodos de manipulação do objeto cliente, enquanto o atributo *idade* tem procedimento simples e completo para obter o valor de um atributo.

### EXEMPLO 4:

#### CLIENTE

nome: string  
endereço: string  
data\_nascimento: date  
renda\_mensal: real, range of (3.000, 100.000)  
profissão : string  
idade: execute (current\_year - year of data\_nascimento)  
período da viagem: two of data  
perfil : list of PREFERÊNCIAS,



## MÉTODOS

inclui\_cliente: procedure()  
mostra\_cliente : procedure()

Utilizando a mesma forma de representação, as informações de tipo texto ou imagem são definidas, elas mesmas, como frames-descrição cujos atributos incluem seu tipo e localização (ou seja, o endereço de onde o arquivo de texto ou imagem se localiza). Entre os métodos definidos para esses objetos incluem-se necessariamente as chamadas aos programas de gerência de imagens e textos.

### 3.1.2 FRAME-REGRA

São formas estruturadas que descrevem a seqüência de *raciocínios* que o sistema realiza sobre as informações que possui e compõem o conhecimento normalmente expresso na forma de regras. Esses objetos permitem que o sistema tome decisões que, em um sistema de banco de dados convencional, seriam unicamente da competência do utilizador. Essa característica permite uma maior independência do sistema e exige do usuário menos conhecimento sobre a forma e conteúdo dos dados. A seguir, são listadas algumas regras informais que um agente de viagens utilizaria para construção de um pacote de viagem.

*Se cliente tem renda individual superior a US\$ 1.500*

*E cliente tem período de férias superior 15 dias*

*Então cliente deseja viagem internacional.*

*Se cliente tem preferência por museus*

*E cliente tem preferência por teatro*

*E cliente tem preferência por arte*

*Então cliente deseja pacote cultural*

*Se cliente planeja viagem internacional*

*E cliente deseja pacote cultural*

*E período do evento coincide com as férias*

*Então destino da viagem = Feira Arte de Paris*

Essas regras são modeladas através de frames-regra como no exemplo 5:

### EXEMPLO 5

#### VIAGEM

identificação: internacional

objetos envolvidos: CLIENTE

disparar quando: renda > 50.000,

essencial: período > 15 dias

#### SUGESTÃO

identificação: cultural

objetos envolvidos: CLIENTE

disparar quando: profissão one of (professor, artista)



essencial: /

assessórios: perfil one of (museu, teatro, arte)

negação: perfil one of (camping, esportes)

#### PACOTE

identificação: Feira de Arte de Paris

objetos envolvidos: CLIENTE, PASSEIO

disparar quando: VIAGEM.tipo = internacional,  
SUGESTÃO.tipo=cultural

essencial: PASSEIO.duração\_média  
< CLIENTE.período\_de\_férias

assessórios: perfil one of (francês, vinhos, gastronomia)

negação: execute(verifica-visto-entrada) = visto\_não\_válido

Genericamente, o frame-regra pode ser descrito como um frame especial cujos atributos correspondem a uma *condição* da regra. Cada condição corresponde a um ou mais pares de atributos e valores referentes aos frames que descrevem os objetos envolvidos na regra (descritos explicitamente através do atributo *objetos-envolvidos*).

Esses pares devem ser comparados com as instâncias dos objetos armazenados no banco de dados. Se os valores casarem, o objeto identificado pelo frame-regra passa a ser verdadeiro para aqueles objetos envolvidos na identificação. A seguir, é descrita a estrutura básica de um frame-regra:

#### NOME DO FRAME

identificação: string

objetos envolvidos: list of OBJETO (qualquer objeto representado no sistema)

disparar quando: list of CONDIÇÃO

essencial: list of CONDIÇÃO;  
list of PROCEDIMENTO

assessórios : list of CONDIÇÃO;

negação: list of CONDIÇÃO;

O atributo "identificação" refere-se a conclusão buscada pela regra. O atributo "objetos envolvidos" lista os objetos cujos atributos serão verificados pela regra. Pode ser qualquer objeto que possua instâncias representadas no sistema.

O atributo "disparar quando" acelera o processo de casamento de padrões, selecionando em primeiro lugar as regras mais prováveis de serem comprovadas para um determinado objeto. No exemplo, para a regra de construção de pacotes, um pacote para a Feira de Arte de Paris será considerado inicialmente para clientes que potencialmente fariam uma viagem internacional e tem predileção por programas culturais.

O atributo "essencial" detalha o casamento de padrões da regra, verificando se o objeto envolvido (no caso cliente) possui todas as premissas para comprovar a regra. No

caso, essas premissas podem ser atributos e valores, ou procedimentos de verificação a serem executados ou ainda, consultas particulares ao usuário, para obter novos dados.

O atributo "assessórios" lista outras informações que reforçariam uma determinada conclusão em detrimento de outra. Ou seja, a conclusão representada no frame é válida mesmo que as informações representadas nos assessórios não sejam comprovadas. Porém, entre duas conclusões válidas (por exemplo, dois possíveis pacotes de viagem) uma pode mostrar-se mais adequada que outra.

O atributo "negação" lista informações que invalidam a conclusão representada no frame. Como o atributo "disparar quando" também auxiliam o processo de seleção de regras a serem verificadas.

### 3.1.3 INFERÊNCIA

O raciocínio do sistema é feito através de uma busca para frente e para trás combinadas. Inicialmente, o sistema tenta casar as características do usuário contra os frames-regra, selecionando os frames-regra candidatos através de uma busca para frente utilizando os atributos *disparar-quando* das regras. A partir da análise destes o sistema busca um pacote turístico que se adequa ao cliente em questão. Para isso, é desenvolvida uma busca para trás ou dirigida por objetivos, onde as informações complementares são solicitadas ao usuário.

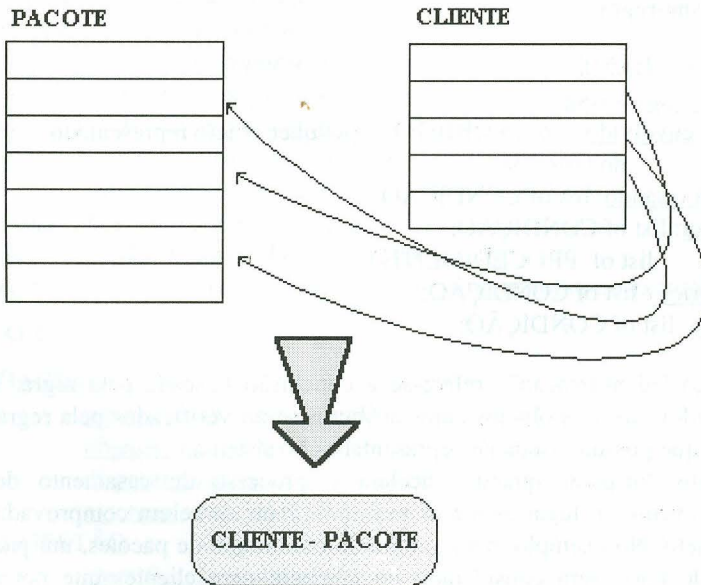




Figura 2 - Através dos valores dos atributos de CLIENTE que são casados contra os atributos de PACOTE pode-se obter o par CLIENTE-PACOTE que satisfaça a consulta.

Se nenhum dos pacotes descritos identifica-se com as necessidades do cliente, o sistema tenta construir um pacote de acordo com suas características. O pacote final é obtido a partir de um raciocínio para frente, ou dirigido por dados. A inferência propriamente dita ocorre por casamento de padrões dos valores de atributos dos diversos objetos envolvidos. Os atributos do tipo *é-um* implementam a herança de atributos e permitem inferir propriedades de frames específicos que são herdadas de frames mais gerais (como no exemplo 2, onde PASSEIO E MUSEU herdam os atributos de EVENTO). Na aplicação da Agência de viagens, o objetivo é construir uma dupla CLIENTE-PACOTE, onde os elementos do pacote serão deduzidos ao longo da interação entre o cliente e a agência de viagens.

### 3.2 A FERRAMENTA DE IA

O segundo elemento chave do sistema, ao lado do modelo conceitual, é a ferramenta de processamento simbólico que permite criar uma representação interna das informações, na máquina fiel ao modelo conceitual descrito.

Para isso é utilizada uma ferramenta construída a partir da Linguagem Lisp

[11] implementando os tipos de frames necessários ao sistema bem como a forma de inferência. A Linguagem Lisp é utilizada por possuir:

- as *listas de propriedades* como primitivas básicas para a construção de frames;
- capacidade de manipular chamadas de procedimentos e funções como objetos, o que possibilita definir procedimentos como atributos de frames;
- poder e flexibilidade para a manipulação simbólica.

A ferramenta implementa a partir das listas de propriedades do Lisp os dois tipos de frames necessários ao sistema (descrição e regra). São definidos os procedimentos para implementar a herança de atributos através dos atributos *é-um* e as ligações semânticas laterais do tipo *associação*, para permitir que as consultas sejam recuperadas em frames relacionados. O motor de inferência implementa as buscas para frente e para trás combinadas, e é dirigido pelas regras-candidatas e pelos atributos de CLIENTE.

As instâncias das informações constantes no banco de dados do sistema são recuperadas, para o casamento contra os frame-regra candidatos, através de consultas feitas na linguagem de manipulação do banco de dados. As consultas são geradas pela ferramenta simbólica como descrito na próxima seção.

### 3.3 O BANCO DE DADOS E SUA INTEGRAÇÃO COM A FERRAMENTA DE PROCESSAMENTO SIMBÓLICO

As informações de grande volume, como as de CLIENTE, EVENTO, etc, são manejadas eficientemente quando estão armazenadas em um Banco de Dados. O acesso e atualização são realizados via linguagem de manipulação de dado suportada pelo Sistema de Gerência de Bancos de Dados (SGBD) correspondente. Esta seção apresenta uma

solução para o problema da representação física das informações de grande volume, baseada na abordagem relacional [3] para bancos de dados.

Na abordagem relacional, os dados são organizados fisicamente como *relações*, que são coleções de *tuplas* ou *registros*, sendo cada tupla composta por uma seqüência de valores de *atributo*. Tuplas são identificadas por valores correspondentes a um atributo ou grupo de atributos, chamados de *chave primária* da relação correspondente. Alguns atributos ou grupos de atributos de uma relação podem ter valores iguais aos de chaves primárias de outra relação. Neste caso, são chamados de *chaves estrangeiras*.

Uma forma simples de associar frames a estruturas de dados relacionais é descrita na Tabela 1. Considera-se aqui como "atributo valorado" aquele cujo valor correspondente é simples, como *integer*, *real*, *string*, *date*, ou valor de uma dada lista (como *tipo de museu* no exemplo 2).

Os valores do tipo texto e imagem são armazenados em arquivos separados (as "bases" de textos e imagens), em formatos próprios para sua manipulação pelos programas específicos. Os atributos calculados, ou que serão obtidos por procedimentos de consulta ao usuário (*execute*), não tem representação armazenada e são gerados no instante da consulta.



Tabela 1 - O mapeamento entre os elementos do modelo de frames para o modelo relacional.

Modelo de Frames	Modelo Relacional
Conjunto de instâncias de um frame.	Relação
Atributos valorados do frame, incluindo <i>one of valor</i> e <i>list of valor</i> .	Atributos de relação
Atributos identificadores do frame.	Chave primária da relação
<i>é-um</i>	Chave estrangeira
Atributo associação ( <i>one of frame</i> -descrição)	Chave estrangeira
Atributo associação ( <i>n of frame</i> -descrição ou <i>list of frame</i> -descrição)	Relação com dois atributos, onde o primeiro é a chave primária da relação correspondente ao frame origem e o segundo é a chave primária da relação correspondente ao frame associado.
Atributos com valores "longos" (textos, imagens)	Atributo da relação que terá como valores apontadores para os bancos de texto e de imagens.

A linguagem de manipulação padrão para a abordagem relacional é a SQL (*Structured Query Language*) [3]. É nessa linguagem que se expressam as requisições de dados necessários aos procedimentos de casamentos de padrões indicados pelos frames-regra. Durante a execução de uma inferência, quando um frame-regra é ativado, diversas consultas ao banco de dados são feitas. Comandos em SQL são preparados de antemão para realizar estas consultas. O conjunto de todos estes comandos caracteriza o "mapeamento" da ferramenta simbólica para o banco de dados, mapeamento este gerado durante a implementação da aplicação específica do sistema. Esta geração é realizada pelo *tradutor de modelos*, usado pelo implementador do sistema.

Como exemplo do resultado do trabalho do tradutor, apresenta-se no exemplo 6 o conjunto de consultas correspondente aos frames-regra VIAGEM, SUGESTÃO E PACOTE, quando um cliente de nome X quer viajar.

## EXEMPLO 6

### Relações necessárias

CLIENTE( nome, endereço, datanasc, renda, profissão, períodoinicio, períodofim)  
 CLIENTE\_PERFIL( clientenome\*, perfilnome\*)

EVENTO(nome, local)  
VISITA\_MUSEU(eventonome\*, tipo, tempovisita)  
PASSEIO(eventonome\*, duraçãomédia)

Obs.: os atributos sublinhados são chaves primárias, os assinalados com "\*" são chaves estrangeiras.

**Consultas**

frame-regra VIAGEM:

```
SELECT CLIENTE.renda,  
       CLIENTE.períodoinicio,  
       CLIENTE.períodofim  
FROM CLIENTE  
WHERE CLIENTE.nome = X
```

frame-regra SUGESTÃO

```
SELECT CLIENTE.profissão,  
       CLIENTE_PERFIL.perfilnome  
FROM CLIENTE, CLIENTE_PERFIL  
WHERE CLIENTE.nome = X  
       AND CLIENTE_PERFIL.clientenome = CLIENTE.nome
```

frame-regra PACOTE

```
(1) SELECT CLIENTE_PERFIL.perfil,  
      duraçãoevento = MUSEU.tempovisita  
FROM CLIENTE, CLIENTE_PERFIL, EVENTO, MUSEU  
WHERE CLIENTE.nome = X  
      AND CLIENTE.nome = CLIENTE_PERFIL.clientenome  
      AND EVENTO.nome = MUSEU.eventonome
```

```
(2) SELECT CLIENTE_PERFIL.perfil,  
      duraçãoevento = PASSEIO.duraçãomédia  
FROM CLIENTE, CLIENTE_PERFIL, EVENTO, PASSEIO  
WHERE CLIENTE.nome = X  
      AND CLIENTE_PERFIL.clientenome = CLIENTE.nome  
      AND EVENTO.nome = PASSEIO.eventonome
```

Este trabalho não aborda os procedimentos de otimização de consultas, que necessariamente deverão ser incluídos para obter um mapeamento eficiente. O estudo destes procedimentos faz parte da continuação do presente trabalho.



## 4 CONCLUSÕES, COMENTÁRIOS FINAIS.

A quantidade de informações existentes, estruturadas e não estruturadas, que precisam ser controladas e manuseadas seletivamente por profissionais de diversos setores econômicos, especialmente por aqueles que realizam tarefas de prospecção e de elaboração de diagnósticos complexos, ultrapassa a capacidade da tecnologia atual para sistemas de informação. Nenhum sistema que se possa desenvolver com os recursos de modelagem e implementação atualmente disponíveis, consegue, isoladamente, dar o apoio mecânico de que os profissionais acima necessitam. Uma das conclusões deste trabalho é a de que um conjunto de técnicas advindas de sistemas diferentes (de base de dados, de recuperação de textos, de base de conhecimentos), integradas sob uma mesma interface inteligente de comunicação com seu utilizador, compõe um Sistema de Informação Total, que pode ser a solução para este problema.

Neste trabalho se discutiu o problema de modelagem conceitual para os SIT, e se propôs um modelo conceitual de informações capaz de representar os requisitos de dados para estes sistemas. Algumas idéias sobre a implementação do sistema, e sobre como mapear o modelo conceitual para as estruturas dos sistemas de gerência de bancos de dados, de textos e imagens, são apresentadas.

As propostas deste trabalho constituem parte de um projeto de pesquisa que tem como objetivos a definição de um método de projeto, desenvolvimento e implementação de SIT's sobre plataformas UNIX, usando ao máximo pacotes de "software" disponíveis. Algumas áreas de aplicação estão sendo consideradas, além da Agência de Viagens: Prospecção Geológica e Apoio à Decisão em Medicina, entre outras.

## REFERÊNCIAS.

- [1] Barr Avron and Feigenbaum, Edward A. (eds). Knowledge representation. in: The handbook of artificial intelligence. Los Altos, Calif., Addison Wesley, 1981. Vol 1 , Chap III.
- [2] Conklin, J. Hypertext: a survey and introduction. IEEE Computer, 20(9), pp 17-41, 1987.
- [3] Date, C.J. An Introduction to Database Systems, Vol. I, Addison Wesley, 1986.
- [4] Diettrich, K.R. Object-oriented database systems: the notion and the issues. In Proceedings of the International Workshop on Object-oriented Database Systems, Pacific Grove, CA, 1986.
- [5] Hayes-Rot, F., Waterman, D., Lenat, D. (Eds). Building Expert Systems, Reading, Massachusetts, Addison Wesley Publ., 1983.
- [6] Parsaye, K., Chignell, M., Khoshafian, S., Wong, H. Intelligent Databases, Toronto, John Wiley & Sons, 1989.
- [7] Intellicorp Inc. KEEconnection: A bridge between databases and knowledge bases. Technical Article. Intellicorp, Menlo Park, California, 1987.

- [8] Mattos, Nelson M. An approach to knowledge base Management. Berlin, Spring-Verlag, 1991.
- [9] Minsky, M. A framework for representing knowledge. In : Winston, D. (Eds). The psychology of computer vision. New York, McGraw Hill, 1975, pp 211-277.
- [10] Salton, G. McGill, M.J. An Introduction to Modern Information Retrieval, New York, McGraw Hill, 1983.
- [11] Winston, Patrick H. and Horn, Berthold K.P. Lisp. Reading, Mass, Addison-Wesley, 1981.

---

<sup>1</sup>Trabalho desenvolvido com auxílio parcial de CNPq e FAPERGS.