UNIVERSIDADE DE SAO PAULO
Instituto de Ciências Matemáticas de São Carlos
Departamento de Ciências de Computação e Estatística

**Armazenagem e Gerenciamento de Informações
Temporais em um Modelo de Bases de Dados
Orientado a Objetos**

Roberto Carlos da Silva

Orientador
Prof.Dr. Caetano Traina Júnior

Dissertação apresentada ao Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo, como parte dos requisitos para a obtenção do Título de Mestre em Ciências - Área: Ciências da Computação e Matemática Computacional.

São Carlos - SP
Brasil
Novembro de 1993

Dedico este trabalho aos meus pais, João e Odite.

AGRADECIMENTOS

Ao Prof. Dr. Caetano Traina Jr., pela orientação neste trabalho, paciência, compreensão e amizade dispensada.

Ao Prof. Dr. Paulo C. Masiero, pelo apoio e colaboração dispensada desde que cheguei ao ICMSC.

Ao Dr. Saulo K. Rodrigues pelo incentivo e apoio, sem o qual este trabalho não se realizaria. Ao Sr. Milton Rocha pelo apoio e incentivo.

Ao Guerino e ao Locoselli, pela amizade, apoio e incentivo.

Aos senhores Vazzoler, Paulo César, e Airton pelo apoio.

Às secretárias da Pós-Graduação, Beth e Laura, e aos funcionários do Laboratório, pela simpatia, atenção e eficiência com que sempre me atenderam.

As funcionárias da Biblioteca, pela colaboração dispensada durante todos os meus estudos realizados no ICMSC.

Aos participantes do Grupo de Estudos de Bases de Dados - ICMSC - USP, Nivaldi, João, Pedro, Valêncio, Takai, Mauro, Rildo e Luiz, que contribuíram nas discussões para o desenvolvimento deste trabalho e a amizade sempre presente. Ao Gustavo, que colaborou para a realização deste trabalho.

A minha irmã Roseli, pela amizade, carinho e apoio.

À CAPES, pelo apoio financeiro parcial dado para o desenvolvimento deste trabalho.

SUMÁRIO

Lista de Figuras.	- viii -
Resumo.	- x -
Abstract.	- xi -
 Capítulo 1. Introdução	 - 1 -
1.1 - Considerações Iniciais	- 1 -
1.2 - Motivação	- 1 -
1.3 - Objetivos	- 3 -
1.4 - Organização da Dissertação	- 4 -
 Capítulo 2. Bases de Dados Temporais	 - 5 -
2.1 - Considerações Iniciais	- 5 -
2.1.1 - Tempo	- 5 -
2.1.2 - Intervalo de Tempo	- 5 -
2.1.3 - Evento	- 5 -
2.1.4 - Conceitos de Tempo	- 6 -
2.2 - Classificação de Bases de Dados com Relação a Tempo	- 6 -
2.2.1 - Base de Dados Instantânea	- 6 -
2.2.2 - Base de Dados "Rollback"	- 7 -
2.2.3 - Base de Dados Histórica	- 7 -
2.2.4 - Base de Dados Temporal	- 8 -
2.3 - Propostas de Tratamento de Tempo em Bases de Dados	- 9 -
2.3.1 - A Linguagem de Consulta Temporal TQuel	- 9 -
2.3.1.1 - Introdução	- 9 -
2.3.1.2 - Definições Formais	- 10 -
2.3.1.2.1 - Definição do Esquema	- 10 -
2.3.1.2.2 - Constantes e Funções Pré-Definidas	- 11 -
2.3.1.2.3 - Tipos de Dados Temporais	- 11 -
2.3.1.3 - Expressões Temporais	- 11 -
2.3.1.3.1 - Construtor Temporal	- 11 -
2.3.1.3.2 - Expressão-E	- 12 -
2.3.1.3.3 - Expressão-I	- 12 -

2.3.1.3.4 - Operador Predicado Temporal	- 12 -
2.3.1.3.5 - Predicado Temporal	- 13 -
2.3.1.4 - Cláusulas	- 13 -
2.3.1.4.1 - A Cláusula "When"	- 13 -
2.3.1.4.2 - A Cláusula "Valid"	- 13 -
2.3.1.4.3 - A Cláusula "As-Of"	- 14 -
2.3.1.5 - Comandos	- 14 -
2.3.1.5.1 - Comando "Retrieve"	- 15 -
2.3.1.5.2 - Comando "Append"	- 16 -
2.3.1.5.3 - Comando "Delete"	- 16 -
2.3.1.5.4 - Comando "Replace"	- 17 -
2.3.1.5.5 - Comando "Create"	- 18 -
2.3.2 - Modelagem Lógica de Dados Temporais	- 18 -
2.3.2.1 - Construções Para Dados Temporais	- 18 -
2.3.2.1.1 - Classes	- 18 -
2.3.2.1.2 - Sequências de Tempo	- 18 -
2.3.2.1.3 - Coleções de Sequências de Tempo	- 18 -
2.3.2.1.4 - TSCs Simples	- 19 -
2.3.2.1.5 - TSCs Complexas	- 19 -
2.3.2.2 - Propriedades Semânticas de Dados Temporais	- 19 -
2.3.2.2.1 - Granularidade	- 19 -
2.3.2.2.2 - Tempo de Vida	- 20 -
2.3.2.2.3 - Regularidade	- 20 -
2.3.2.2.4 - Densidade	- 20 -
2.3.2.3 - Operações em TSCs	- 20 -
2.3.2.3.1 - Seleção	- 21 -
2.3.2.3.2 - Agregado	- 22 -
2.3.2.4 - Mapeando o Modelo de Dados Temporal no Modelo Relacional	- 22 -
2.3.2.4.1 - A Necessidade de Relações Temporais	- 22 -
2.3.2.4.2 - A Representação Tabular Escolhida	- 22 -
2.3.2.4.3 - Forma Normal Temporal	- 23 -
2.3.2.4.4 - Representação Relacional de Uma TSC	- 23 -
2.3.2.4.5 - Família de TSCs	- 23 -
2.3.3 - A Linguagem de Consulta Gráfica Time-by-Example (TBE)	- 24 -
2.3.3.1 - Definições	- 24 -
2.3.3.2 - Especificação de Consulta	- 25 -
2.3.3.2.1 - Esqueleto de Relação e Variáveis	- 25 -
2.3.3.2.2 - Caixa de Saída	- 25 -
2.3.3.2.3 - Caixa de Condição	- 25 -
2.3.3.2.4 - Caixa de Faixa de Valores	- 25 -
2.3.3.3 - Exemplo	- 25 -
2.3.4 - Semântica Formal Para Tempo em Bases de Dados	- 26 -
2.3.5 - Modelo de Dados Orientado Temporalmente	- 27 -

2.3.6 - Modelo Relacional Homogêneo Para Bases de Dados Temporais ..	- 27 -
2.3.7 - Projetando Suporte Temporal Para SGBD	- 28 -
2.3.8 - Sumário de Projetos	- 28 -
2.3.8.1 - Projeto de Sistemas de Gerenciamento Baseado em Conhecimento e Problemas Orientado ao Tempo	- 28 -
2.3.8.2 - Estrutura Unificada Para Projeto e Gerenciamento de Bases de Dados Temporais	- 28 -
2.3.8.3 - O Projeto do SGBD POSTGRES	- 29 -
2.4 - Como Embutir Relações Temporais em Relações Instantâneas	- 29 -
2.5 - Considerações Finais	- 30 -
Capítulo 3. Características de Tempo	- 31 -
3.1 - Considerações Iniciais	- 31 -
3.2 - Tipo de Base de Dados em Relação à Classificação Adotada	- 32 -
3.3 - Modelo de Dados Básico	- 32 -
3.4 - Modelo Conceitual Temporal	- 33 -
3.5 - Desenvolvimento de um Modelo Para Bases de Dados Temporais	- 33 -
3.6 - Linguagem de Consulta Temporal	- 33 -
3.7 - Atributo/Objeto "Time-Stamp"	- 34 -
3.8 - Ponto/Intervalo de Tempo	- 35 -
3.9 - Propriedades Semânticas	- 35 -
3.10 - Implementação	- 35 -
3.11 - Aplicação Alvo	- 36 -
3.12 - Considerações Finais	- 36 -
Capítulo 4. O Modelo de Representação de Objetos	- 38 -
4.1 - Considerações Iniciais	- 38 -

4.2 - Conceituação	- 38 -
4.2.1 - Esquema de Dados Utilizado nos Exemplos	- 38 -
4.2.2 - Diagramas de Representação	- 38 -
4.2.3 - Fundamentos	- 40 -
4.2.4 - Esquema e Meta-Esquema de Dados	- 42 -
4.3 - GERenciador de Objetos - GEO	- 42 -
4.4 - Considerações Finais	- 45 -
Capítulo 5. Conceitos Temporais No MRO*	- 46 -
5.1 - Considerações Iniciais	- 46 -
5.2 - Conceitos Temporais	- 47 -
5.2.1 - Dimensões Temporais	- 47 -
5.2.2 - Atributo/Objeto Temporizado	- 48 -
5.2.3 - Intervalo/Ponto de Tempo	- 49 -
5.2.4 - Semântica Temporal	- 49 -
5.3 - Como os Conceitos Temporais Afetam o MRO*	- 49 -
5.3.1 - Atributos Temporais	- 49 -
5.3.2 - Atributo Temporizado	- 51 -
5.3.3 - Intervalo de Tempo	- 52 -
5.3.4 - Propriedades Semânticas	- 52 -
5.3.4.1 - Granularidade	- 52 -
5.3.4.2 - Tempo de Vida	- 53 -
5.3.4.3 - Densidade	- 54 -
5.3.5 - Considerações Sobre Propriedades Semânticas no MRO*	- 54 -
5.3.6 - Definição de Atributo Temporal no Esquema	- 56 -
5.3.7 - Exemplo de Definição de Informações Temporais	- 57 -
5.4 - Determinação de Valores de um Atributo Temporizado	- 57 -
5.5 - Considerações Finais	- 58 -
Capítulo 6. LAMRO*	- 59 -
6.1 - Considerações Iniciais	- 59 -
6.2 - Especificações	- 59 -
6.2.1 - Classificação dos Comandos	- 59 -

	- v -
6.2.2 - Notação Utilizada	- 60 -
6.3 - Exemplos da Linguagem	- 61 -
6.3.1 - Operações de Elementos do MRO* no Esquema	- 61 -
6.3.1.1 - Inclusão de Objeto	- 62 -
6.3.1.2 - Inclusão de Atributo	- 62 -
6.3.1.3 - Eliminação de Atributo	- 63 -
6.3.2 - Operações de Elementos do MRO* na Base de Dados	- 63 -
6.3.2.1 - Inclusão de Objetos	- 63 -
6.3.2.2 - Inclusão de Valor de Atributo	- 64 -
6.3.2.3 - Eliminação de Valor de Atributo	- 64 -
6.4 - Extensão da Linguagem Temporal	- 64 -
6.4.1 - Operações de Elementos do MRO* no Esquema	- 64 -
6.4.1.1 - Inclusão de Atributo Temporal	- 64 -
6.4.1.2 - Inclusão de Atributo Temporizado	- 65 -
6.4.1.3 - Eliminação de Atributo Temporal	- 67 -
6.4.1.4 - Eliminação de Atributo Temporizado	- 67 -
6.4.2 - Operações de Elementos do MRO* na Base de Dados	- 68 -
6.4.2.1 - Inclusão de Valor de Atributo Temporizado	- 68 -
6.4.2.2 - Eliminação de Valor de Atributo Temporizado	- 69 -
6.4.3 - Operações do Tempo de Vida	- 70 -
6.4.3.1 - Inclusão de Valores do Tempo de Vida	- 70 -
6.4.3.2 - Substituição de Valores do Tempo de Vida	- 70 -
6.5 - Substituição	- 70 -
6.6 - Recuperação	- 71 -
6.6.1 - Objetos	- 71 -
6.6.1.1 - Com relação ao Esquema	- 71 -
6.6.1.2 - Com relação a Base	- 71 -
6.6.2 - Relacionamentos	- 72 -
6.6.2.1 - Com relação ao Esquema	- 72 -
6.6.2.2 - Com relação a Base	- 73 -
6.6.3 - Atributos	- 73 -
6.6.3.1 - Com relação ao Esquema	- 73 -
6.6.3.2 - Com relação a Base	- 73 -
6.7 - Comando LISTA	- 74 -
6.7.1 - Cláusula QUANDO	- 75 -
6.8 - Estabelecimento de Ambiente	- 75 -
6.8.1 - Estabelecimento de Sessão	- 75 -
6.8.2 - Valores do Ambiente	- 76 -

6.8.3 - Ambiente Não Estabelecido	- 77 -
6.8.3.1 - Inclusão	- 77 -
6.8.3.2 - Consulta	- 77 -
6.8.3.3 - Exclusão	- 77 -
6.9 - Considerações Finais	- 77 -
Capítulo 7. Descrição da Implementação	- 79 -
7.1 - Considerações Iniciais	- 79 -
7.2 - Descrição dos Arquivos Lógicos	- 79 -
7.2.1 - Estrutura de Dados Básica	- 81 -
7.2.2 - Arquivo Lógico dos Componentes do MRO* Afetados Pelo Tempo	- 81 -
7.2.3 - Arquivo Lógico de Propriedade	- 81 -
7.2.4 - Arquivo Lógico de Tempo	- 82 -
7.2.5 - Estrutura de Dados do Ambiente	- 82 -
7.3 - Rotinas Básicas de Acesso à Tempo	- 83 -
7.3.1 - Rotinas de Acesso ao Esquema	- 83 -
7.3.2 - Rotinas de Acesso a Base de Dados	- 84 -
7.4 - Impacto nas Rotinas de Propriedade	- 85 -
7.4.1 - Rotinas de Propriedades Impactadas	- 85 -
7.5 - Considerações Finais	- 86 -
Capítulo 8. Conclusões e Sugestões de Futuras Pesquisas	- 88 -
8.1 - Considerações Iniciais	- 88 -
8.2 - Decisões de Projeto	- 88 -
8.2.1 - Estrutura de Dados	- 88 -
8.2.2 - Propriedades Semânticas	- 89 -
8.3 - Sugestões de Futuras Pesquisas	- 89 -
8.3.1 - Colônias	- 89 -
8.3.2 - Características do MRO*	- 90 -
8.3.3 - Linguagem de Manipulação dos Dados	- 90 -
8.3.4 - Evolução de Esquema	- 91 -
8.3.5 - Manutenção de Bases de Dados	- 91 -
8.3.6 - Armazenamento de Dados	- 91 -

8.3.7 - Granularidade	- 92 -
8.4 - Considerações Finais	- 92 -

Referências Bibliográficas.	- 95 -
---	---------------

Apêndice A - Exemplo de Utilização da LAMRO.

Apêndice B - Rotinas de Manipulação de Tempo.

LISTA DE FIGURAS

Figura 2.1 - Relação Instantânea.	6
Figura 2.2 - Relação Rollback.	7
Figura 2.3 - Relação Histórica.	8
Figura 2.4 - Relação Temporal	8
Figura 2.5 - Relações Docente e Submetido.	9
Figura 2.6 - Relação Associados	10
Figura 2.7 - Relação Associados Estendida	11
Figura 2.8 - Relações Docente e Submetido Estendidas.	12
Figura 2.9 - TSCs Exemplo	21
Figura 2.10 - Exemplo da Linguagem TBE.	26
Tabela 3.1 - Sumário da Análise dos Modelos.	37
Figura 4.1 - DRO da Aplicação de Informações Financeiras.	39
Figura 4.2 - DRI da Aplicação de Informações Financeiras.	41
Figura 4.3 - Meta-Esquema de Dados do MRO*	43
Figura 4.4 - Estrutura do GEO	45
Figura 5.1 - Dimensões Temporais Agindo em um Atributo	49
Figura 5.2 - DRO com Definição de Atributos Temporizados	50
Figura 5.3 - DRI com Definição de Atributos Temporizados	51
Figura 5.4 - Componentes do MRO* Afetados.	52
Figura 5.5 - Meta-Esquema do MRO* Temporal	53

Figura 5.6 - Relacionamento Entre Propriedades Semânticas e Atributos Temporais. . .	55
Figura 5.7 - Caminho que Determina os Valores de um Atributo Temporizado	57
Figura 5.8 - Valor de um Atributo Temporizado por Vários Atributos Temporais	58
Figura 6.1 - Valores Temporais do Ambiente	77
Figura 7.1 - Arquivo de Objetos.	79
Figura 7.2 - Arquitetura do GEO.	80
Figura 7.3 - Arquivo de Relacionamento	81
Figura 7.4 - Arquivo de Relacionamento Triplo.	82
Figura 7.5 - Arquivo de Propriedade.	82
Figura 8.1 - Estrutura de Dados Adotada Inicialmente	89
Tabela 8.1 - Tabela 3.1 Ampliada Incluindo Comparação com GEO Temporal.	94

ARMAZENAGEM E GERENCIAMENTO DE INFORMAÇÕES TEMPORAIS EM UM MODELO DE BASES DE DADOS ORIENTADO A OBJETOS

Autor: **ROBERTO CARLOS DA SILVA**
Orientador: **Prof. Dr. CAETANO TRAINA JÚNIOR**

RESUMO

Neste trabalho foi desenvolvido um **Modelo de Dados Temporal** que agrega conceitos dos modelos de dados pesquisados na literatura sobre tratamento de **tempo** em Modelos de Dados e Bases de Dados conhecidas como **Bases de Dados Temporais (BDT)** - sendo mostrado como pode ser realizada a incorporação desse modelo em um **Modelo Orientado a Objetos**. O Modelo de Dados Temporal desenvolvido proporciona suporte genérico ao tratamento de tempo, permitindo que estruturas semânticas sejam temporizadas.

Além disso, é descrita a implementação desse Modelo de Dados Temporal que propicia suporte a tempo de forma eficiente. Também foi desenvolvida uma Linguagem de Consulta necessária para a realização das operações com a Base de Dados.

STORAGE AND MANAGEMENT OF TEMPORAL INFORMATION IN AN OBJECT ORIENTED DATABASE MODEL

Author: **ROBERTO CARLOS DA SILVA**

Adviser: **Prof. Dr. CAETANO TRAINA JÚNIOR**

ABSTRACT

This work is concerned with the development of a **Temporal Data Model** based on concepts found in the literature, on the treatment of **time** in **Data Models** and **Databases** known in the literature as **Temporal Databases (TDB)**. It is shown how this Model can be incorporated into an **Object Oriented Model**. This Model provides generic capabilities for handling of time, allowing that semantic structures be temporized.

Furthermore, the implementation of this Temporal Model is described.

A Query Language was also developed which permits operations involving time and Databases.

Capítulo 1

Introdução

1.1 - Considerações Iniciais.

O Tempo é uma parte essencial das informações em função da constante evolução envolvendo o mundo real. Os fatos necessitam ser interpretados no contexto do tempo. Relacionamentos entre eventos ou entidades estão embutidos na informação temporal. Tempo e memória são conceitos inseparáveis, e o estudo de modelos de dados e Sistemas de Gerenciamento de Bases de Dados (SGBDs) que capturem e preservem a dinâmica inerente do conteúdo de Bases de Dados (BD) é um dos maiores impulsos na pesquisa sobre o tempo e sistemas de informação. A intensificação da atividade de pesquisa em torno do tempo e de BD é incentivada pela observação de que as atividades humanas estão inseridas no tempo, porém, é difícil formular explicitamente este contexto temporal.

1.2 - Motivação.

A premissa básica da expansão da pesquisa neste assunto, é o reconhecimento de que tempo não é meramente outra dimensão, ou qualquer outro atributo em um objeto, mas algo que os seres humanos tratam de modo especial. Os resultados destas pesquisas asseguram a percepção que o projeto de características temporais em sistemas de informação requer novas e diferentes ferramentas conceituais e práticas. O Tempo é uma dimensão universal de aplicações de gerenciamento de dados, e deste modo o foco das atenções neste domínio tem sido a conceitualização e projeto de um sistema de propósito geral que realize o tratamento da dinâmica dos dados de um modo explícito.

Modelos de dados e SGBDs convencionais supostamente modelam a realidade, mas, a menos de extensões que foram estudadas, tais como [ARIA86], [SNOD87], e [SEGE88], não possuem capacidade para registrar e processar aspectos dependentes de tempo. Atualmente em sistemas de informação, aspectos de tempo são algumas vezes negligenciados, outros tratados apenas implicitamente, ou explicitamente fatorados do sistema. Em geral, sistemas de informação e ferramentas de gerenciamento de dados não tratam simetricamente dados atuais e passados, mas os diferenciam em termos de disponibilidade para acesso lógico e físico.

A ausência de suporte a informações de tempo obriga as aplicações mais sofisticadas que utilizam SGBDs a manter e manipular informações temporais de uma maneira própria. Por exemplo, SGBDs convencionais não suportam consultas históricas sobre estados passados (que são importantes para sistemas de apoio a decisão). Não há maneiras de se representar mudanças retroativas e/ou proativas. Aplicações voltadas para experimentos científicos e estatísticos também necessitam de suporte a dados temporais, onde experimentos físicos, medidas, simulações, e coleta de estatísticas entre outros, são dependentes de tempo.

Aplicações comerciais em geral necessitam acessar apenas dados correntes. Aplicações

em bases de dados científicas e estatísticas (SSDBs - "Statistical and Scientific Databases") são inerentemente dependentes de tempo. Entretanto, dados temporais também são essenciais em aplicações comerciais. Em várias aplicações tais como sistemas de controle de bancos, vendas, controle de inventário e sistemas de reservas mantêm-se um histórico completo das transações na BD. Frequentemente esse histórico necessita ser analisado estatisticamente para propósitos de tomada de decisão. Outras aplicações onde o tratamento de informações temporais é inerente, são BD de engenharia, econometria, biblioteconomia, análise política, música, medicina, legislação, automação de escritórios e industrial.

Os SGBDs atuais tratam atributos de tempo como quaisquer outros atributos. Eles não têm qualquer entendimento da semântica de tempo, ou das propriedades de dados temporais. Atualmente, os usuários de bases de dados devem desenvolver e programar sua própria semântica de tempo sobre a aplicação. O principal objetivo de um SGBD temporal é absorver a semântica de tempo, e as características de dados temporais, no próprio SGBD. Consequentemente, os usuários estariam livres da tarefa de transformar a semântica e características temporais em processos computacionais, e poderiam utilizar diretamente os recursos proporcionados pelo SGBD temporal.

Diversos esquemas têm sido propostos para suportar processamento de informação temporal em SGBDs incorporando um ou mais atributos de tempo. Algumas destas propostas são abordadas neste trabalho. Bolour em [BOLO82] apresenta um "survey" bibliográfico contendo várias referências a pesquisas sobre tempo e BD. McKenzie em [MCKE86] apresenta uma extensa bibliografia sobre o tratamento de tempo em modelos de dados e SGBD. Snodgrass em [SNOD90] apresenta um sumário dos dez artigos e eventos, que considera mais importante em termos de impacto em BD temporais, e também o estado atual e direções futuras em BD temporais. Soo em [SOO*91] apresenta uma bibliografia em que constam artigos que foram publicados a partir de 1988, e artigos que não apareceram nos levantamentos bibliográficos anteriores. Soo afirma que a sua bibliografia sobre tempo em BD é a quarta a ser publicada. Ela é um complemento à bibliografia de Stam e Snodgrass [STAM88], que complementa o "survey" de McKenzie [MCKE86], que por sua vez complementa o "survey" de Bolour e outros [BOLO82].

A capacidade de Bases de Dados Temporais¹ (BDT), de modo geral, é armazenar e gerenciar todos os dados relativos a uma aplicação [ELMA91]. Isto inclui o estado corrente da aplicação, tanto quanto um "histórico completo" de todas as mudanças ocorridas na base, incluindo o momento em que ocorrem. Adicionalmente, estados futuros da base são armazenados e gerenciados, podendo ser consultados e atualizados.

Afora a complexidade conceitual envolvida, o esforço para implementar BDT sofre da ausência de meios adequados para manipular o grande volume de dados a serem armazenados, necessários para manter um "histórico completo". A relevância prática de modelos de dados temporais é deste modo dependente dos desenvolvimentos recentes de hardware que tornam as restrições tradicionais para armazenamento de grandes volumes de dados menos rígidas. Neste sentido uma tendência promissora é o desenvolvimento de dispositivos ópticos para armazenamento de massa. Portanto, antes que a capacitação e

¹Bases de Dados Temporais são tratadas no capítulo 2.

promessas de BDTs sejam alcançadas, há numerosos problemas técnicos que devem ser analisados e resolvidos.

O **primeiro desafio** é projetar modelos de dados e linguagens de consulta que permitam aos usuários projetar, implementar, consultar e atualizar BDTs. O modelo de dados e a linguagem de consulta devem proporcionar conceitos e construções que facilitem a representação, consulta e gerenciamento de dados temporais. Dentro desta perspectiva, a área de Orientação a Objetos deve contribuir para a solução deste problema. Um modelo de dados orientado a objetos deve incorporar o máximo possível da semântica das aplicações. Tempo é sem dúvida um conceito que permeia grande parte dos sistemas de software e, por outro lado, é passível de um tratamento uniforme que atenda a um amplo subconjunto das necessidades de tais sistemas. Assim, é importante a modelagem dos aspectos temporais em sistemas de bases de dados orientados a objetos (SGBDOO).

O **segundo desafio** é desenvolver técnicas para armazenamento e pesquisa de grandes quantidades de dados que devem ser armazenados e gerenciados em um sistema de BDT. Técnicas de pesquisa para dados temporais, tanto quanto combinações de atributos temporais e não temporais são necessárias para realizar implementações eficientes de BDT. A utilização de novas tecnologias, tais como, discos ópticos, ou uma combinação de discos magnéticos e discos ópticos é uma das melhores alternativas para manter BDTs "on-line" e acessíveis [COPE82], [ELMA91].

A classificação proposta por Snodgrass em [SNOD86a] para BDT, representa uma maneira como pode ser feito o tratamento a atributos de tempo, sendo esta a classificação na qual está baseado este trabalho.

Esta classificação baseia-se apenas em atributos temporais, sendo ortogonal aos demais conceitos de um modelo orientado a objetos. A distinção entre os tipos de BDs é feita pela sua habilidade para suportar conceitos de tempo e processamento de informação temporal. Esta classificação permite consolidar a terminologia e evitar confusões comumente encontradas na literatura.

Existe alguma discussão em relação à melhor maneira de representar tempo em aplicações de bases de dados. Uma possibilidade é representar somente **tempo absoluto**, usando alguma granularidade de tempo (dias, horas, minutos, etc.) apropriada, baseando-se nas necessidades da aplicação. Outra possibilidade é representar **tempo relativo** entre eventos. Neste caso, relacionamentos temporais tais como "before", "after", e "during" devem ser expressos sem referência a um tempo absoluto.

1.3 - Objetivos.

Baseado no **Modelo de Representação de Objetos - MRO*** [TRAI86][TRAI88][BLAJ92] está em desenvolvimento no ICMSC - USP São Carlos um Sistema de Gerenciamento de Banco de Dados denominado **GErenciador de Objetos - GEO** [TRAI91], com o objetivo de suportar aplicações de projetos de engenharia e de desenvolvimento de sistemas.

Considerado um modelo orientado a objetos, o MRO* permite representar Colônias, Objetos, Relacionamentos e Atributos, possibilitando incorporar informações semânticas em entidades do mundo real. Neste trabalho definiu-se uma maneira como o MRO* trata de

aspectos temporais, o que é feito seguindo a classificação de [SNOD86a]. Tornou-se disponível através de atributos o tratamento de informações temporais para objetos e relacionamentos.

A temporização da BD propicia maiores facilidades para o usuário desenvolvedor de aplicativos, pois todo o tratamento de tempo é realizado automaticamente pelo SGBD, e para o usuário final, pois todas as informações pertinentes a um objeto do mundo real estão armazenadas na própria representação do objeto, não sendo necessária a criação de históricos das informações de objetos. Para atingir estes objetivos, neste trabalho são estudados os critérios para a temporização de uma BD. Colocado em prática, este trabalho incorporou primitivas ao GEO para permitir a temporização de uma BD.

1.4 - Organização da Dissertação.

O presente trabalho está organizado em oito capítulos, sendo que o restante desta dissertação se encontra organizado da seguinte maneira:

Capítulo 2 : Apresenta-se a classificação de BDTs adotada e uma revisão bibliográfica com propostas de tratamento de tempo em modelos de dados.

Capítulo 3 : Um conjunto de características de tratamento de tempo em modelos de dados é apresentado. Algumas das propostas apresentadas no capítulo dois são analisadas de acordo com este conjunto.

Capítulo 4 : É apresentado o modelo de dados (MRO*) onde se desenvolveu este trabalho, estabelecendo-se alguns de seus conceitos e também o sistema (GEO) onde foram obtidos os resultados aqui apresentados.

Capítulo 5 : Descreve-se a extensão realizada no MRO* para incorporar conceitos temporais.

Capítulo 6 : É apresentada a Linguagem de Acesso ao MRO* - LAMRO, em conjunto com as alterações acrescentadas para atender os requisitos temporais.

Capítulo 7 : São descritos alguns aspectos da implementação realizada neste trabalho.

Capítulo 8 : São apresentadas as conclusões deste trabalho, sugestões para futuras pesquisas e decisões de projeto.

Capítulo 2

Bases de Dados Temporais

2.1 - Considerações Iniciais.

Neste capítulo inicialmente são apresentados alguns conceitos de tempo e a proposta de classificação de BDT formulada por Snodgrass [SNOD86a].

Em seguida discorre-se sobre trabalhos disponíveis na literatura que visam incorporar o tratamento de informações de tempo em modelos de dados e SGBD.

2.1.1 - Tempo.

O tempo varia continuamente e pode ser representado por números reais. Contudo, neste trabalho considera-se **tempo** como uma sequência de pontos de tempo equidistantes, isto é, 0,1,2,3, ..., "now", onde o símbolo 0 representa o tempo de início relativo e "now" é o tempo (momento) presente. Entre dois pontos de tempo consecutivos t_i e t_{i+1} há um período de tempo que é equivalente a uma unidade de tempo ou seja, para qualquer ponto de tempo tem-se $t_i = t_{i-1} + 1$. Este período é indivisível, a menos que uma unidade menor de tempo seja utilizada. A unidade de tempo não é pré-determinada e cabe ao usuário da base de dados a sua especificação. Essa especificação é feita pelo usuário quando da definição da propriedade semântica granularidade².

2.1.2 - Intervalo de Tempo.

Um **intervalo** de tempo é um conjunto de pontos de tempo consecutivos e representa um período de tempo. O intervalo é representado por um limite inferior (i) e um limite superior (s), cuja representação é $[i,s)$. Este intervalo inclui i , todos os pontos entre i e s , não inclui s , isto é este intervalo é aberto no limite superior. O intervalo que inclui "now" como seu limite superior, $[i,now]$ é um caso especial. Ele é fechado tanto no limite inferior quanto no superior. Este tipo de intervalo expande-se com a evolução do tempo, sendo diferente de outros tipos de intervalo nos quais a duração é fixa. O símbolo ∞ (infinito) indica que o limite superior é infinito, ou seja não existe limite superior.

2.1.3 - Evento.

Um **evento** é uma ocorrência instantânea que representa algum fato, ação ou condição sendo satisfeita. Eventos significativos em BD, são transações, operações ou ações que

²Esta propriedade é a que está definida em [SEGE87], e o enquadramento desta propriedade neste trabalho é apresentado na seção 5.3.4.1.

causam o assinalamento de novos valores aos atributos dos objetos da BD [TANS86]. A representação de um evento que ocorra em um instante t pode ser tanto $[t, t]$ como $[t, t + 1]$. A diferença é que em $[t, t]$ o intervalo é fechado no limite inferior e no superior, e em $[t, t + 1)$ o intervalo é fechado no limite inferior e aberto no superior.

2.1.4 - Conceitos de Tempo.

Os três conceitos de tempo (tempo da transação, tempo válido e tempo definido pelo usuário) que são fundamentais para a classificação que foi adotada neste trabalho são abordados nesta seção. Existem várias dimensões de tempo em BDs [SNOD86a]. Neste trabalho segue-se a abordagem utilizada por Snodgrass, onde **tempo da transação** (ou tempo físico) é o momento em que a informação é armazenada na base de dados, **tempo válido** (ou tempo lógico) é o momento a partir do qual a informação modelada é válida. Para usuários em que o tempo da transação e tempo válido não satisfaçam suas necessidades, permite-se que uma dimensão denominada **tempo definido pelo usuário** seja definida. Essa dimensão de tempo tem sua semântica definida e interpretada por funções definidas pelo usuário e não pelo SGBD temporal. Os atributos temporais definidos pelo usuário não são interpretados pelo SGBD, e apenas a representação interna e funções de entrada e saída são necessárias.

2.2 - Classificação de Bases de Dados com Relação a Tempo.

Snodgrass propõe uma classificação de tipos de BDs, diferenciadas pelas suas habilidades para representar informações temporais em [SNOD86a]. Ressalte-se que os argumentos utilizados em [SNOD86a] baseiam-se no modelo relacional. Nas próximas seções são apresentados os tipos de BD segundo a classificação adotada.

2.2.1 - Base de Dados Instantânea.

Bases de dados convencionais modelam a dinâmica do mundo real como um instantâneo em um ponto particular do tempo. Um estado ou uma instância de uma BD é seu conteúdo corrente que não reflete necessariamente a realidade do mundo real. O estado de uma BD é atualizado utilizando-se operações de manipulação de dados, tais como, inserção, eliminação e substituição, tendo efeito tão logo as operações terminem. Desta maneira, dados antigos da base são descartados e esquecidos. A este tipo de BD, denomina-se **base de dados instantânea**.

Este tipo de BD é inadequado em muitas situações, tais como, consultas históricas, análise de tendências, mudanças retroativas e/ou proativas.

No modelo relacional, uma BD instantânea é uma coleção de relações. Cada relação consiste de um conjunto de tuplas

Figura 2.1 - Relação Instantânea

e é usualmente representada como uma tabela bi-dimensional. Essas relações serão denominadas relações instantâneas. A figura 2.1 ilustra esta situação.

2.2.2 - Base de Dados "Rollback".

Uma abordagem para resolver parte das deficiências apresentadas pelas BD instantâneas é armazenar todos os estados passados, indexados pelo tempo. Tal abordagem requer a representação do **tempo da transação**, que indica o momento em que a informação é armazenada na base. Esta representação pode ser considerada como uma sequência de estados instantâneos indexados pelo tempo. Desta maneira, é possível que deslocando-se pelo eixo do tempo, seja selecionado um estado instantâneo, possibilitando consultas sobre este estado. A operação de seleção de um estado instantâneo é conhecida como "rollback" e uma BD que a suporte é conhecida como **base de dados "rollback"**.

Uma relação nesta abordagem pode ser ilustrada conceitualmente em três dimensões (figura 2.2), com o tempo da transação servindo como terceiro eixo. A relação pode ser considerada como uma sequência de relações instantâneas indexadas pelo tempo da transação.

Uma característica semântica importante de dados temporais é que para um dado objeto, os valores do tempo da transação são totalmente ordenados no tempo.

Mudanças em uma BD "rollback" só podem ser executadas no estado mais recente. Cada transação resulta em um novo estado sendo adicionado, avançando o eixo do tempo. Deve-se observar contudo que uma das limitações de propiciar suporte a tempo da transação é que fica registrado o histórico das atividades da BD, e não as do mundo real.

Não há modos de se corrigir erros passados. A única possibilidade é corrigir erros no estado corrente, porém, os erros permanecerão registrados no estado anterior. Também não é possível registrar mudanças retroativas/proativas.

Figura 2.2 - Relação "Rollback"

2.2.3 - Base de Dados Histórica.

Bases de dados históricas registram um único estado histórico por relação, enquanto que BD "rollback" registram uma sequência de estados instantâneos. Em BDs históricas cada informação registrada é mantida junto com a informação do instante a partir do qual aquela informação é válida.

Quando erros são descobertos, eles são corrigidos através de modificações na base. Não são mantidos registros de estados anteriores da base, nem de erros corrigidos. Logo, BD

históricas representam conhecimento corrente sobre estados passados, enquanto que BD "rollback" podem retornar a estados anteriores corretos ou incorretos, sem no entanto poder alterá-los.

Bases de dados históricas propiciam suporte para **tempo válido**, que é o momento a partir do qual a informação modelada é válida, ou seja, o tempo associado com a perspectiva da aplicação do usuário. A semântica de tempo válido é muito próxima a realidade, por consequência é mais complexa do que a semântica de tempo da transação, que diz respeito as atividades da base.

Bases de dados históricas podem ser ilustradas em três dimensões. Neste caso, o terceiro eixo é representado pelo tempo válido. A figura 2.3 ilustra esta situação.

Figura 2.3 - Relação Histórica

2.2.4 - Base de Dados Temporal.

Esta abordagem incorpora características tanto de BD históricas quanto de BD "rollback", proporcionando suporte tanto para **tempo da transação** quanto para **tempo válido**.

Segundo Snodgrass, o termo base de dados temporal foi utilizado para dar ênfase à necessidade de se proporcionar suporte tanto para tempo da transação quanto para tempo válido na manipulação de informações temporais. Uma operação "rollback" em uma relação temporal, seleciona um estado histórico, no qual podem ser executadas consultas, atualizações, etc. Cada transação causa a criação de um novo estado histórico. Em decorrência do conceito de que em BD temporais as informações não são eliminadas, tem-se que relações temporais são apenas acrescentadas ("append-only").

Este tipo de BD pode ser ilustrado em quatro dimensões, visto que dois eixos referentes ao tempo (tempo válido e tempo da

Figura 2.4 - Relação Temporal

transação) estão envolvidos. A figura 2.4 (sequência a, b, c, d) ilustra esta situação.

2.3 - Propostas de Tratamento de Tempo em Bases de Dados.

2.3.1 - A Linguagem de Consulta Temporal TQuel.

A linguagem TQuel foi proposta em [SNOD87] com o intuito de realizar operações em uma BDT. A linguagem TQuel é derivada da linguagem de consulta Quel, do SGBD relacional INGRES. Segundo o autor, a linguagem TQuel foi projetada para ser uma extensão mínima, tanto sintática como semântica, da linguagem Quel. A abordagem utilizada pelo autor foi a de adicionar novos atributos para representar tempo no modelo relacional para a representação de BDTs. Como o modelo relacional não incorpora tempo diretamente, a linguagem de consulta TQuel deve traduzir as consultas e atualizações envolvendo tempo em recuperações e modificações nas relações instantâneas.

A seguir, são abordados alguns conceitos da linguagem de consulta TQuel.

2.3.1.1 - Introdução.

Será utilizada a base de dados com as relações Docente e Submetido, para exemplificar as consultas com TQuel. A relação Docente lista os docentes membros de uma faculdade e suas posições (os valores são Assistente, Associado e Titular). A relação Submetido lista os artigos submetidos para publicação. A figura 2.5 ilustra este exemplo que foi extraído do artigo [SNOD87].

O comando "retrieve" da linguagem Quel consiste de dois componentes básicos: a lista alvo especificando como os atributos da relação sendo derivada são computados a partir dos atributos das relações básicas, e uma cláusula "where", especificando quais as tuplas que participam na derivação.

O comando "range" associa tuplas com relações; esta vinculação permanece até que um novo comando "range" seja executado com a mesma tupla.

Exemplo : Listar os professores associados.

```
range of f is Docente
retrieve into Associados (Nome
= f.Nome)
where f.posição = "Associado"
```

Docente(Nome,Posição):	Nome	Posição
	Jane	Titular
	Maria	Associado
	Tom	Associado
Submetido(Autor,Jornal):	Autor	Jornal
	Jane	CACM
	Maria	CACM
	Maria	TODS
	Tom	JACM

Figura 2.5 - Relações Docente e Submetido

A consulta anterior produz a relação Associados apresentada na figura 2.6. Esta relação será utilizada nos próximos exemplos.

A abordagem de [SNOD87] quanto a relações temporais é embutí-las em uma relação instantânea, adicionando dois atributos temporais. O valor do primeiro atributo especifica o tempo válido. Para a relação evento, que consiste de tuplas representando ocorrências instantâneas, este atributo contém um único valor de tempo ("at"). Para a relação intervalo, que consiste de tuplas representando um estado válido em um intervalo de tempo, o atributo contém dois valores de tempo delimitando o intervalo ("from, to"). O segundo atributo especifica o tempo da transação, i. é, o momento quando a informação foi armazenada na base. Dois valores de tempo estão sempre associados com o tempo da transação : o momento em que a tupla foi armazenada na base ("start"), e o momento em que ela foi removida ("stop"). Na figura 2.7 apresenta-se a relação Associados estendida. Na figura 2.8 apresenta-se a relação Docente estendida para tornar-se uma relação intervalo, e a relação Submetido estendida para tornar-se uma relação evento. O símbolo ∞ (infinito) foi utilizado para registrar o tempo em que o argumento "stop" não existe. A granularidade dos valores de tempo válido é arbitrária; para simplificação, assume-se que ela é de um mês/ano.

2.3.1.2 - Definições Formais.

2.3.1.2.1 - Definição do Esquema.

O comando "create" define uma relação e proporciona um esquema para aquela relação. O comando **create persistent interval** Docente (Nome = c20, Posição = c10) define a relação Docente referenciada anteriormente. As palavras chaves "persistent", "interval" ou "event" são extensões relativas à linguagem TQuel, sendo opcionais. A palavra "persistent" refere-se a relações "rollback" ou temporal. As palavras "interval" ou "event" referem-se a relações históricas ou temporais. Entretanto, se nenhuma das palavras chaves anteriormente mencionadas são utilizadas, tem-se uma relação instantânea convencional. Logo, os quatro tipos de relações (instantânea, "rollback", histórica e temporal) podem ser especificadas. A especificação do domínio de dados é similar àquela da linguagem Quel (são permitidos inteiros, números de ponto flutuante, cadeias de caracteres), com a adição de tipos de dados temporais.

Um par de valores de tempo da transação, "start" e "stop", está associado a relações "rollback" e temporais. Quanto a relações históricas e temporais, tem-se duas situações. Quando trata-se de um evento tem-se um único tempo válido, "at". Para intervalo, tem-se associado um par de tempo válido, "from" e "to".

Associados(Nome):	Nome
	Maria
	Tom

Figura 2.6 - Relação Associados

2.3.1.2.2 - Constantes e Funções Pré-Definidas.

A linguagem Quel suporta constantes numéricas e cadeias de caracteres. A linguagem TQuel possibilita trabalhar também com constantes temporais. As cadeias de caracteres que são utilizadas com as cláusulas "valid", "when", "as-of" são interpretadas como constantes temporais denotando um intervalo de tempo particular. Por exemplo, a cadeia de caracter "Setembro 1, 1983" denota um intervalo de 00:00 hs. do dia 1 de Setembro de 1983 até 23:59 hs. do dia 1 de Setembro de 1983; "Setembro de 1983" denota todo o mês. As constantes "beginning", "now" e "forever" estão disponíveis para serem utilizadas com tempo válido e tempo da transação.

Os atributos temporais implícitos estão disponíveis através das funções "validat", "validfrom", e "validto" (para tempo válido), e "transactionstart" e "transactionstop" (para tempo da transação), para uso somente em lista alvo e cláusula "where".

Atributos definidos pelo usuário não podem ser utilizados em cláusulas subsequentes "when", "valid", ou "as-of".

Associados(Nome):				
	Tempo válido		Tempo da transação	
Nome	(from)	(to)	(start)	(stop)
Jane	12-76	11-80	09-83	
Maria	12-82		09-83	
Tom	12-80		09-83	

Figura 2.7 - Relação Associados Estendida

2.3.1.2.3 - Tipos de Dados Temporais.

A linguagem TQuel provê um tipo de dado temporal para proporcionar suporte a tempo definido pelo usuário. Como mencionado anteriormente, os atributos temporais definidos pelo usuário não são interpretados pelo SGBD temporal; somente a representação interna e as funções de entrada e saída são fornecidas.

2.3.1.3 - Expressões Temporais.

2.3.1.3.1 - Construtor Temporal.

Um construtor temporal é um operador unário ou binário que utiliza-se de um ou dois eventos ou intervalos como argumentos, e retorna um evento ou intervalo. O prefixo unário dos construtores temporais são "begin of" e "end of", ambos retornando eventos. Os operadores infix binário dos construtores temporais são "overlap" e "extend", ambos retornando intervalos. O operador "overlap" é indefinido se não há valores de tempo que estão em ambos os intervalos básicos. A idéia que o operador "overlap" transmite é semelhante a um operador de intersecção temporal, em que ele retorna os pontos no tempo quando ambos os argumentos são válidos. O operador "extend" é semelhante a uma união

temporal, em que ele retorna os pontos no tempo em que ambos os argumentos são válidos.

2.3.1.3.2 - Expressão-E.

Uma expressão-e ("e-expression") é uma expressão contendo tuplas, constantes temporais, e construtores temporais, devendo a expressão resultar em um evento. Expressões-e são usadas nas cláusulas "valid" e "as-of". Expressões-e devem ter "begin of" ou "end of" como operadores de mais alto nível.

2.3.1.3.3 - Expressão-I.

Uma expressão-i ("i-expression") é uma expressão contendo tuplas, constantes temporais e construtores temporais que evoluem para um intervalo.

Docente(Nome,Posição):					
		Tempo válido		Tempo da transação	
Nome	Posição	(from)	(to)	(start)	(stop)
Jane	Assistente	09-71	12-76	09-71	
Jane	Associado	12-76	11-80	12-76	
Jane	Titular	11-80		10-80	∞
Maria	Assistente	09-77	12-82	08-77	
Maria	Associado	12-82		12-82	∞
Tom	Associado	09-75		08-75	10-75
Tom	Assistente	09-75	12-80	10-75	
Tom	Associado	12-80		11-80	∞
Submetido(Autor,Jornal):					
		Tempo válido		Tempo da Transação	
Autor	Jornal	(at)	(start)	(stop)	
Jane	CACM	11-79	11-79	∞	
Maria	CACM	09-78	09-78	∞	
Maria	TODS	05-79	05-79	∞	
Tom	JACM	12-82	12-82	∞	

Figura 2.8 - Relações Docente e Submetido Estendidas

2.3.1.3.4 - Operador Predicado Temporal.

O **operador predicado temporal** é um operador infixo binário que tem como argumentos, eventos e intervalos, resultando em um valor Booleano. Os três operadores são "**precede**", "**overlap**", "**equal**". O operador "overlap" é um operador sobrecarregado semanticamente, já que ele é tanto um construtor temporal, como também um predicado temporal. O operador "overlap" especifica que os eventos e/ou intervalos sobrepõem-se no tempo.

2.3.1.3.5 - Predicado Temporal.

Um **predicado temporal** é uma expressão contendo operadores lógicos ("and, or, not") operando em expressões contendo um operador predicado temporal, que por sua vez operam expressões-e e expressões-i.

Um operador predicado temporal mapeia pares de intervalos ou eventos para uma variável Booleana, que pode ser operada por operadores lógicos. Predicados temporais são usados apenas em cláusulas "when".

2.3.1.4 - Cláusulas.

A seguir, apresentam-se as cláusulas de manipulação dos atributos temporais.

2.3.1.4.1 - A Cláusula "When".

A cláusula "when" é a análoga temporal a cláusula "where" da linguagem Quel. Esta cláusula consiste da palavra chave seguida pelo predicado temporal nas tuplas, representando os atributos de tempo implícito das relações associadas.

Ex.: Liste os professores associados em setembro de 1983.

```
range of a is Associados
retrieve into PrimeiroDiaAssoc(Nome = a.Nome)
  when a overlap "09-83"
```

O resultado desta consulta é Maria e Tom.

Quando utiliza-se intervalos de tempo, os dois valores de tempo do atributo implícito que representam o mesmo (início - "start" e fim - "stop") podem ser indicados por operadores unários "begin of" e "end of".

2.3.1.4.2 - A Cláusula "Valid".

A cláusula "valid" especifica o tempo durante o qual a tupla derivada é válida.

Há duas variantes para esta cláusula. No caso de intervalos, utiliza-se a variante "valid from .. to ..". Para eventos, utiliza-se a variante "valid at" para especificar o valor do tempo no atributo temporal.

Ex.: Quando os professores associados foram promovidos para esta posição ?

```
range of a is Associados
retrieve into PromAssociado (Nome = a.Nome)
  valid at begin of a
```

Jane foi promovida em dezembro de 1976, Maria em dezembro de 1982 e Tom em dezembro de 1980. Nesta consulta a relação básica, Associados, é uma relação intervalo. O argumento "start" foi selecionado como o valor do tempo na relação (evento) derivada.

Ex.: Quem obteve promoção de assistente para titular enquanto ao menos um outro membro permaneceu como associado ?

```
range of f1 is Docente
range of f2 is Docente
range of a is Associados
retrieve into Talentos (Nome = f1.Nome)
  valid from begin of f1 to begin of f2
  where f1.Nome = f2.Nome and f1.Posição = "Assistente" and
    f2.Posição = "Titular"
  when (f1 overlap a) and (f2 overlap a)
```

Tuplas na relação derivada Talentos indicam o intervalo de tempo no qual os professores associaram-se à faculdade como assistentes para tornar-se titulares.

2.3.1.4.3 - A Cláusula "As-Of".

As cláusulas "when" e "valid" são usadas para expressar consultas históricas. Para consultas "rollback", utiliza-se a cláusula "as-of".

Ex.: Utilizando-se a relação Talentos do exemplo anterior, deseja-se saber quais eram os professores que satisfaziam tais condições no final de 1984 ?

```
range of f1 is Docente
range of f2 is Docente
range of a is Associados
retrieve into Talento84 (Nome = f1.Nome)
  valid from begin of f1 to begin of f2
  where f1.Nome = f2.Nome and f1.Posição = "Assistente" and
    f2.Posição = "Titular"
  when (f1 overlap a) and (f2 overlap a)
  as of end of "1984"
```

A cláusula "as-of" retorna ao estado em que a base de dados se encontrava em 31 de dezembro de 1984, e avalia o restante da consulta usando a informação conhecida somente naquele momento. Adições de informações ou correções de erros feitas depois daquele momento não estão incluídas na relação resultante.

2.3.1.5 - Comandos.

Dos 14 comandos da linguagem Quel, 5 destes comandos foram estendidos na linguagem TQuel. São os comandos "create", "retrieve", "append", "delete" e "replace".

A linguagem Quel possui três comandos de atualização : "append", "delete", e "replace".

Estes comandos na linguagem TQuel não possuem cláusula "as-of", pois, o tempo da transação é computado automaticamente pelo SGBD temporal como o tempo corrente (BDTs são "append only"). Contudo, as cláusulas "valid" e "when" podem ser empregadas nestes comandos.

2.3.1.5.1 - Comando "Retrieve".

Para manipular os atributos temporais, o comando "retrieve" da linguagem Quel foi estendido com três componentes, um especificando como o atributo temporal válido é computado, e dois especificando o relacionamento temporal das tuplas participando na derivação. Os componentes são as cláusulas "valid", "when" e "as-of".

A sintaxe para o comando "retrieve" é³ :

range of t_1 is R_1

...

range of t_k is R_k

retrieve ($t_{i_1}.D_{j_1}, \dots, t_{i_r}.D_{j_r}$)

valid from v to χ

where ψ

when τ

as of α through β

onde : R_k são as relações,

t_k são as tuplas,

$t_{i_r}.D_{j_r}$ são os atributos das tuplas,

v e χ indicam o tempo válido que restringe as tuplas,

ψ é uma expressão do cálculo de predicados de primeira-ordem,

τ é o predicado temporal,

α e β definem uma sequência de transações.

As relações resultantes após a execução de um comando "retrieve" são fundidas quando elas são armazenadas.

³As variáveis utilizadas no comando "Retrieve" serão utilizadas nos comandos seguintes.

2.3.1.5.2 - Comando "Append".

A sintaxe para o comando "append" é :

range of t_1 is R_1

...

range of t_k is R_k

append $(t_{i_1}.D_{j_1}, \dots, t_{i_r}.D_{j_r})$

valid from v to χ

where ψ

when τ

O conjunto a ser incorporado na relação deve conter somente tuplas que não estejam armazenadas na relação.

Exemplo : A professora Maria foi contratada um mês antes do que fôra registrado anteriormente.

```
append to Docente (Nome = "Maria", Posição = "Assistente")  
  valid from "8-77" to "12-82"
```

A tupla que será adicionada terá o argumento "valid to" com o valor 9-77, pois, já existe uma tupla na relação cujos valores para "valid from" e "valid to" são respectivamente 9-77 e 12-82.

2.3.1.5.3 - Comando "Delete".

A sintaxe para o comando "delete" é :

range of t_1 is R_1

...

range of t_k is R_k

range of s is R

delete s

valid from v to χ

where ψ

when τ

O comando "delete" muda o valor do argumento "stop" do tempo da transação de ∞ (*infinito*) para "now" se as cláusulas "when" e "where" forem válidas em algum momento, ocorrendo também a adição de tuplas para as quais atribui-se o valor "now" ao argumento "start" do tempo da transação, se qualquer uma das tuplas a serem eliminadas não cobrirem completamente uma tupla existente.

Exemplo : A professora Jane desligou-se do departamento em Março de 1988.

```
range of f is Docente
delete f
  where f.Nome = "Jane"
  valid from "3-88"
```

Ao argumento "stop" no tempo da transação da última tupla que estava associada a Jane será atribuído o valor "now", que indica o momento em que o comando "delete" é executado. Em seguida, tem-se a adição de uma nova tupla em que os valores dos argumentos "from", "to" (tempo válido), "start" e "stop" (tempo da transação) serão respectivamente o valor de "valid from" da tupla anterior, 3-88, "now" e ∞ (*infinito*) .

2.3.1.5.4 - Comando "Replace".

A sintaxe para o comando "replace" é :

```
range of  $t_1$  is  $R_1$ 
...
range of  $t_k$  is  $R_k$ 
range of s is R
replace s ( $t_{i_1}.D_{j_1}, \dots, t_{i_r}.D_{j_r}$ )
  valid from  $v$  to  $\chi$ 
  where  $\psi$ 
  when  $\tau$ 
  as of  $\alpha$  through  $\beta$ 
```

Exemplo : Em outubro de 1985 soube-se que Tom tinha submetido um artigo não para JACM, mas para TOPLAS, um mês depois do que supunha-se anteriormente.

```
range of s is Submetido
replace s (Jornal="TOPLAS")
  where s.Autor = "Tom" and s.Jornal = "JACM"
  valid at begin of "Janeiro 1983"
```

2.3.1.5.5 - Comando "Create".

Este comando foi abordado na seção 2.3.1.2.1.

2.3.2 - Modelagem Lógica de Dados Temporais.

A abordagem de A. Shoshani e A. Segev em [SEGE87] e [SEGE88] é examinar a semântica e desenvolver construções para dados temporais de maneira independente de qualquer modelo de dados (tais como relacional, entidade-relacionamento).

A seguir, abordam-se alguns conceitos apresentados em [SEGE87] e [SEGE88].

2.3.2.1 - Construções Para Dados Temporais.

As construções básicas do modelo de dados temporal (TDM - "Temporal Data Model") são definidas nesta seção.

2.3.2.1.1 - Classes.

Uma classe é qualquer coleção de objetos que tenham os mesmos atributos. Todo objeto de uma classe tem um identificador único, denominado "surrogate". Uma classe pode ter um ou mais identificadores. Uma classe composta é uma classe que possui mais do que um identificador.

2.3.2.1.2 - Sequências de Tempo.

Um valor de dado temporal é uma tripla $\langle s, t, a \rangle$, onde s é o "surrogate", t é o tempo e a é o atributo. Para dados não temporais, t é considerado o valor corrente e omitido.

Uma característica desses dados é que para um dado "surrogate" e um determinado atributo, os valores de dados temporais estão totalmente ordenados no tempo, isto é, formam uma sequência ordenada. Essas sequências são denominadas sequências de tempo (TS - "Time Sequences").

2.3.2.1.3 - Coleções de Sequências de Tempo.

Coleção de sequência de tempo (TSC - "Time Sequence Collection") é uma coleção de TSs para a mesma classe de "surrogate" e com as mesmas propriedades.

TSCs são utilizadas para expressar os atributos temporais de uma classe e relacioná-los com outros atributos (temporais ou não) da classe.

2.3.2.1.4 - TSCs Simples.

TSC simples é definida para uma classe simples (uma classe com identificador único) e um único atributo temporal. Essas TSCs podem ser vistas como a coleção de todos os valores temporais de um único atributo para todos os "surrogates" de uma classe simples.

2.3.2.1.5 - TSCs Complexas.

TSCs complexas são TSCs cujos componentes S, T e A não representam um único elemento.

O caso em que S não é um único elemento corresponde a uma classe composta, e é representado por (S,T,A). Essa estrutura é útil na representação do comportamento temporal de relacionamentos e seus atributos.

O caso em que T não é um único elemento corresponde à situação em que valores temporais têm mais do que uma TS associada a eles, e é representado por (S,T,A). Essa situação está presente nos conceitos de tempo válido, tempo da transação e tempo definido pelo usuário definidos no trabalho de Snodgrass, vide [SNOD86a].

O caso em que A não é um único elemento, ocorre quando muitos atributos ocorrem nos mesmos instantes, e é representado por (S,T,A).

É possível a ocorrência simultânea dos três casos apresentados.

2.3.2.2 - Propriedades Semânticas de Dados Temporais.

Nesta seção descrevem-se as propriedades semânticas de dados temporais.

Os autores apresentam uma distinção entre pontos de tempo ("time points") e pontos de dados ("data points") de uma TS. Pontos de tempo são todos os pontos no tempo que podem assumir valores de dados. Pontos de dados são os pontos que possuem valores de dados associados com eles.

A seguir, apresentam-se as propriedades semânticas de TSs que são : granularidade, tempo de vida, regularidade e densidade ("type").

2.3.2.2.1 - Granularidade.

Esta propriedade especifica a granularidade dos pontos de tempo de uma TS, i. é, os pontos no tempo que podem ter valores de dados.

A representação da granularidade do tempo pode ser feita de duas maneiras: **ordinal** e **calendário**. A representação ordinal implica que os pontos de tempo são contados pela

posição ordinal inteira (1,2,3,...). A representação calendário pode assumir os valores de tempo de um calendário, tais como, ano, mês, dia, hora, etc.

2.3.2.2.2 - Tempo de Vida.

Esta propriedade representa o **tempo de vida** associado a cada TS. O tempo de vida é especificado por um ponto inicial ("start-point") e um ponto final ("end-point") definindo a extensão dos pontos de tempo válidos para a TS. A representação dos tempos inicial ("start-time") e final ("end-time") pode ser tanto ordinal como calendário.

2.3.2.2.3 - Regularidade.

Uma TS é denominada **regular** se contém um valor para cada ponto de tempo no intervalo do tempo de vida associado a ela. Logo, os pontos de dados para uma TS regular são os mesmos pontos de tempo da TS. Por outro lado, uma TS é denominada **irregular** se contém valores apenas para um subconjunto próprio dos pontos de tempo associados ao tempo de vida da TS.

2.3.2.2.4 - Densidade.

Esta propriedade determina os valores da TS para pontos de tempo que não possuem valores explícitos. Em geral, há uma função de interpolação associada com cada TS. A propriedade densidade aplica-se a TSs regulares e irregulares, e possui as seguintes variantes: passo constante ("step-wise constant"), contínua, discreta e definida pelo usuário. Os atributos a_i e t_i apresentados a seguir significam, respectivamente, atributo e tempo.

A densidade **passo constante** é definida por: se (t_i, a_i) e (t_k, a_k) são dois pares consecutivos na TS, tal que $t_i < t_k$, então $a_j = a_i$ para $t_i \leq t_j < t_k$.

A densidade **contínua** é definida por: uma função contínua é assumida entre (t_i, a_i) e (t_k, a_k) que assinala a_j para t_j ($t_i \leq t_j \leq t_k$) baseada em uma função de assentamento de curva.

A densidade **discreta** é definida por: cada valor (a_i) na TS não está relacionado com outros valores. Consequentemente, valores inexistentes não podem ser interpolados.

A densidade **definida pelo usuário** é definida por: valores inexistentes na TS podem ser calculados com base em funções de interpolação definidas pelo usuário.

2.3.2.3 - Operações em TSCs.

Os operadores apresentados nesta seção obedecem à dois princípios. O primeiro princípio é que todo operador executado em uma ou mais TSCs, terá como resultado uma

única TSC. O segundo princípio é que todo operador deve ter três partes funcionais: especificação alvo, mapeamento e função. Este princípio assegura a consistência dos operadores.

A **especificação alvo** determina os pontos válidos da TSC resultante, que é o resultado de uma operação entre TSCs.

A parte **mapeamento** especifica para cada ponto da TSC resultante, o conjunto de pontos da TSC fonte a ser utilizado para gerar um valor temporal.

A parte **função** especifica a função a ser aplicada aos pontos das TSCs fonte para gerar o resultado final.

A seguir apresentam-se alguns operadores de recuperação do TDM. A sintaxe adotada é baseada em uma estrutura semilar à linguagem SQL ("Structured Query Language") e sua forma geral é :

```
nome-operador INTO tsc-alvo função
FROM tsc-fonte [,tsc-fonte]*
WHERE especificação alvo
GROUP BY | TO especificação-de-mapeamento
```

onde : | denota ou, [] denota expressões opcionais, e * denota zero ou mais ocorrências.

Os exemplos que serão utilizados na apresentação de cada operador, utilizam as TSCs apresentadas na figura 2.9. Os exemplos são os mesmos apresentados em [SEGE87]. Para a TSC vendas_livro o atributo temporal quantidade contém o número de livros vendidos por dia.

2.3.2.3.1 - Seleção.

Este operador extrai partes de uma TSC que satisfaçam um predicado, referenciando os valores de s ("surrogate") e/ou t (tempo) e/ou a (atributo).

Todas as propriedades da TSC fonte são herdadas pela TSC resultante, exceto o tempo de vida, que pode ser modificado por uma operação temporal.

Exemplo : Obtenha as vendas de janeiro para os livros número 5 e 9.

```
SELECT INTO VENDAS JAN QUANTIDADE
```

TSC	densidade	descrição	atributo temporal
vendas_livro	discreta	vendas diárias de livros	quantidade
preço_livro	passo constante	preço diário de livros	preço
livro	não temporal	-----	-----
comissão_e mp	discreta	comissão diária do empregado	comissão
salário_emp	passo constante	salário mensal do empregado	salário

Figura 2.9 - TSCs exemplo

```
FROM VENDAS_LIVRO  
WHERE S IN (5,9) AND T IN (1/1/91 TO 1/31/91)
```

As propriedades da TSC resultante são :

granularidade - um dia

densidade - discreta

tempo de vida - mês de janeiro

2.3.2.3.2 - Agregado.

Este operador aplica-se a grupos, tendo como variantes o tempo ou "surrogate".

Todas as propriedades da TSC fonte são herdadas pela TSC resultante, exceto a granularidade que pode ser modificada e a densidade que é alterada para discreta.

Exemplo : Somar as vendas de livros por mês.

```
AGGREGATE INTO VENDAS_MÊS  
SUM QUANTIDADE  
FROM VENDAS_LIVRO  
GROUP T BY MONTH
```

2.3.2.4 - Mapeando o Modelo de Dados Temporal no Modelo Relacional.

Como o intuito desta seção é o mapeamento do modelo TDM no modelo relacional, uma das primeiras questões a serem levantadas é se as estruturas de dados do modelo relacional são poderosas o suficiente para poder representar o modelo TDM. Por exemplo, TSCs podem ser representadas por relações ? Ou como TSCs serão representadas no modelo relacional ?

Para realizar o mapeamento do modelo TDM para o modelo relacional, inicialmente é introduzido o conceito de relações temporais. Em seguida, uma forma normal temporal (TNF), e a representação de uma TSC no modelo relacional.

2.3.2.4.1 - A Necessidade de Relações Temporais.

Pretende-se que os principais conceitos de uma TSC, tais como, densidade, tempo de vida, etc, sejam conceitos de meta-dados que venham a ser incorporados ao modelo relacional. A intenção é estender o modelo relacional incorporando relações temporais que tenham as propriedades semânticas de TSCs associadas a elas.

2.3.2.4.2 - A Representação Tabular Escolhida.

Existem diferentes representações tabulares utilizadas por vários pesquisadores. No

entanto, todas elas podem ser resumidas em quatro categorias, dependendo de dois fatores que são : se as representações tratam com pontos ou intervalos de tempo; e se utilizam a Primeira Forma Normal (FNF - "First Normal Form") ou Não Primeira Forma Normal (NFNF - "Non-First Normal Form").

A escolha dos autores para a representação tabular foi a de utilizar a Primeira Forma Normal (FNF) e a manipulação de pontos de tempo.

A Primeira Forma Normal foi escolhida para representar uma TSC devido o desejo dos autores de permanecer tão próximo quanto possível do modelo relacional.

A utilização da representação de pontos de tempo foi escolhida por ser considerada mais conveniente para dados de eventos. Intervalos de tempo são considerados mais convenientes para dados cuja densidade seja passo constante.

2.3.2.4.3 - Forma Normal Temporal.

Os autores definem uma forma normal temporal denominada primeira forma normal temporal, que será referenciada por 1TNF ("First Temporal Normal Form").

A seguir, apresenta-se a definição da 1TNF quando da utilização de pontos de tempo para a representação tabular.

Definição : Uma relação temporal com um esquema $R(S, T, A_1, \dots, A_n)$ está na 1TNF se não existem duas tuplas $r^1(s^1, t^1, a_1^1, \dots, a_n^1)$ e $r^2(s^2, t^2, a_1^2, \dots, a_n^2)$ tal que $s^1 = s^2$ e $t^1 = t^2$.

Em outras palavras, a 1TNF significa que para uma ocorrência de um "surrogate" em um momento no tempo ("time point"), cada atributo têm um único valor definido.

A seguir, apresenta-se a definição da 1TNF quando da utilização de intervalos de tempo para a representação tabular.

Definição : Uma relação temporal com um esquema $R(S, A_1, \dots, A_n, T_s, T_e)$ está na 1TNF se não existem duas tuplas $r^1(s^1, a_1^1, \dots, a_n^1, t_s^1, t_e^1)$ e $r^2(s^2, a_1^2, \dots, a_n^2, t_s^2, t_e^2)$ tal que $s^1 = s^2$ e os intervalos $[t_s^1, t_e^1]$ e $[t_s^2, t_e^2]$ interseccionem-se.

2.3.2.4.4 - Representação Relacional de Uma TSC.

Representa-se uma TSC por uma relação temporal com um esquema $R(S, T, A)$; a chave da relação é $\langle S, T \rangle$, e a ordenação é primária pelo "surrogate" e secundária por tempo. O que caracteriza a relação temporal, são os meta-dados associados a ela. Os meta-dados são : granularidade, tempo de vida, densidade e regularidade.

2.3.2.4.5 - Família de TSCs.

Em função da escolha da Primeira Forma Normal para a representação tabular, os

autores constatarem que é grande o número de TSCs geradas para um único "surrogate" quando este tem muitos atributos temporais. Esta constatação motivou os autores a introduzirem o conceito de família de TSCs, que é um mecanismo para representar todos os dados de um "surrogate" como uma unidade.

2.3.3 - A Linguagem de Consulta Gráfica Time-by-Example (TBE).

Tansel e outros propõem em [TANS89a] uma linguagem de consulta gráfica que propõe-se ser "amigável ao usuário" denominada Time-by-Example (TBE) projetada especificamente para bases de dados relacionais históricas (HRDB - "Historical Relational Database"). A linguagem TBE segue o conceito de consulta por exemplo, a abordagem gráfica e bi-dimensional da linguagem Query-by-Example (QBE) [ZLOO77] e [DATE86], e emprega a disposição hierárquica de sub-consultas (hierarquia de janelas) da linguagem Summary-Table-by-Example (STBE) [OZSO84].

O modelo de dados utilizado pelo TBE é o modelo relacional com extensões. Para realizar a representação tabular utiliza-se Não Primeira Forma Normal (NFNF)⁴.

TBE é uma linguagem de consulta baseada no cálculo relacional. O modelo de dados permite uma visão tri-dimensional de relações históricas ao nível do usuário, pois, as tabelas são bi-dimensionais e a representação do tempo completa a terceira dimensão. Atributos dependentes e independentes do tempo podem coexistir na mesma relação. É proposta uma álgebra relacional histórica que permite otimização no processamento de consultas no TBE.

Segundo os autores, a principal contribuição da proposta da linguagem TBE é mostrar que é possível projetar uma linguagem de consulta gráfica poderosa e "amigável ao usuário" para bases de dados históricas dentro da estrutura relacional.

2.3.3.1 - Definições.

Seja U o conjunto de valores considerados como elementares (tais como, reais, inteiros, etc.). Um atributo A é monovalorado (atômico) se $D_m \subset U$, onde D_m é o domínio do atributo A . Se $D_m \subset P(U)$ (onde P é o conjunto potência), então A é um atributo multivalorado ("set-valued").

Uma tripla é uma 3-upla cujos componentes são os limites inferior e superior de um intervalo de tempo e um valor válido neste intervalo. Uma tripla é usada como um valor "time-stamp" de um atributo dependente do tempo. Atributos multivalorados dependentes do tempo são modelados por conjuntos de triplas.

⁴Os valores dos atributos permitem repetição, mas, devem ser atômicos.

2.3.3.2 - Especificação de Consulta.

Na linguagem TBE o usuário especifica uma consulta colocando elementos-exemplo no esqueleto da relação fornecida na tela. A consulta é apresentada como uma hierarquia de janelas (sub-consultas). Uma janela consiste de um nome, uma saída, um ou mais esqueletos de relações, e opcionalmente uma caixa de condição e uma caixa de extensão. O usuário comunica-se com o gerente da tela usando o cursor e as teclas de funções do teclado para montar construções na tela e abrir janelas. O nível mais alto de sub-consulta é denominado RAIZ ("ROOT"). Uma consulta TBE consiste de uma raiz e seus descendentes.

2.3.3.2.1 - Esqueleto de Relação e Variáveis.

Um esqueleto de relação representa o esquema da relação. Ele possui colunas para cada atributo da relação e um cabeçalho com o nome da relação e os nomes dos atributos. Possui também linhas para a especificação de tuplas-exemplo de uma consulta.

2.3.3.2.2 - Caixa de Saída.

A saída da raiz ou de uma consulta é um esqueleto de relação sem o cabeçalho. O esqueleto de saída especifica o formato e o conteúdo do resultado da consulta.

2.3.3.2.3 - Caixa de Condição.

A caixa de condição é um esqueleto com uma coluna e algumas linhas. Cada linha contém uma expressão Booleana que consiste de termos conectados por operadores lógicos **not**, **and**, **or**. Nessas linhas o usuário define quais condições devem ser satisfeitas pelas tuplas das relações alvo.

2.3.3.2.4 - Caixa de Faixa de Valores.

A caixa de faixa de valores é um esqueleto com duas caixas e algumas linhas, e é usada para definir relações dinamicamente no domínio de variáveis. A primeira caixa contém variáveis separadas por vírgulas, e delimitadas por parênteses. A segunda caixa contém janelas ou nomes de relações unidas pelo operador "or", e define o domínio das variáveis das relações que estão na primeira caixa de faixa de valores.

2.3.3.3 - Exemplo.

A base de dados utilizada como exemplo será composta pelas seguintes relações:

DIVISÃO(divnome,*\$gerente,localização)
 DEPARTAMENTO(dnome,divisão,*\$gerente,\$orçamento)
 EMPREGADO(enome,\$salário,\$departamento).

Figura 2.10 - Exemplo da Linguagem TBE.

O exemplo da figura 2.10, obtém as divisões nas quais Tom trabalhou de 1975 a 1985.⁵

2.3.4 - Semântica Formal Para Tempo em Bases de Dados.

O modelo para Base de Dados Histórica (HDB - "Historical Database") proposto por J. Clifford e D.S. Warren em [CLIF83] visa proporcionar uma base semântica rigorosa para BD históricas. Isso é realizado através de uma visão linguística de BD, explorando a analogia entre consultas HDB e operações de lógica intencional.

O trabalho de Clifford e Warren foi um dos primeiros a tentar incorporar o conceito de

⁵Na figura 2.10 a utilização dos símbolos "*", "\$" e "\$\$" referem-se as regras de compatibilidade/imcompatibilidade de variáveis, e essas são apresentadas em [TANS89a].

tempo em BD. Seu modelo lógico para a representação de informação temporal é muito pesado, porém a linguagem de consulta é poderosa. Os autores formulam uma lógica intencional na qual baseiam a linguagem de consulta proposta.

O modelo aborda apenas uma dimensão de tempo. Tempo é capturado no modelo através de um conjunto de instâncias de relações "time-stamp". Cada uma dessas instâncias de relações é completa, de modo que todo objeto que já existiu é representado. A relação histórica que é a construção de dados básica no modelo HDB, é definida como a união formal de todas as instâncias completas. Uma BD relacional histórica é uma coleção de relações históricas sobre o mesmo conjunto de estados.

O modelo HDB utiliza alguns atributos internos como suporte para incorporar semântica temporal explícita para o modelo, isto é, esquemas de relação incluem dois atributos implícitos que são: um atributo "time-stamp" que marca o conjunto de todas as tuplas em uma relação, e um indicador booleano para determinar a existência de um objeto.

Dentro da classificação adotada neste trabalho, a proposta de Clifford e Warren é classificada como base de dados histórica.

2.3.5 - Modelo de Dados Orientado Temporalmente.

Gad Ariav em [ARIA86] propõe um modelo denominado TODM "Temporally Oriented Data Model", cujo objetivo é realizar o tratamento de informações temporais no modelo relacional.

Dentre as construções do modelo, destacam-se aquelas que são denominadas esquemas temporais e cubos de dados. Cubos de dados são construções de dados temporais tri-dimensionais onde tempo, objetos e atributos são dimensões primárias do dado armazenado. Enquanto que esquemas de relações determinam a estrutura de um objeto, esquemas temporais introduzem a dimensão do tempo.

A noção de cubo de dados do modelo adiciona profundidade histórica a noção tabular de dados e fornece uma estrutura para armazenamento e recuperação de dados dentro do contexto temporal. Segundo o autor, as operações básicas do modelo permitem a formação de novas visões de cubos existentes.

O modelo TODM define um embasamento semântico para a implementação de um SGBD temporal e conduz ao desenvolvimento de uma linguagem de consulta, TOSQL, que é uma extensão da linguagem de consulta SQL.

Dentro da classificação adotada neste trabalho, a proposta de Ariav é classificada como base de dados temporal.

2.3.6 - Modelo Relacional Homogêneo Para Bases de Dados Temporais.

O modelo proposto por S. Gadia em [GADI88], sugere uma parametrização temporal de relações estáticas, ou seja, considera uma BDT como um modelo para BD relacionais estáticas variando no tempo.

O autor define o conceito de homogeneidade, cujo significado é que os períodos de

validade de todos os atributos em uma tupla da relação temporal são idênticos. Nesse contexto, o autor afirma que seu modelo é homogêneo.

É proposta uma álgebra relacional para o modelo de dados temporal homogêneo e um cálculo de tuplas, e prova-se a equivalência entre ambos.

Dentro da classificação adotada neste trabalho, a proposta de Gadia é classificada como base de dados histórica.

2.3.7 - Projetando Suporte Temporal Para SGBD.

Lum e outros em [LUM*84], abordam os aspectos de projeto e implementação de apenas um aspecto de tempo, o tempo físico no ambiente de um SGBD relacional.

Quanto a representação de tempo, os autores destacam as diferenças entre tempo físico e tempo lógico, e a importância de se representar esses dois conceitos de tempo. Mencionam também que em todas as publicações encontradas (até a publicação da proposta apresentada em [LUM*84]) onde o tempo era tratado, assumia-se que o tempo possui uma única dimensão.

Os autores propõem usar o tempo físico para registrar todas as ações na BD e deixam aos usuários a tarefa de definir explicitamente na BD um ou mais parâmetros de tempo que se julgar necessário.

Dentro da classificação adotada neste trabalho, a proposta de Lum e outros é classificada como base de dados "rollback".

2.3.8 - Sumário de Projetos.

Snodgrass em [SNOD86b] apresenta as linhas de pesquisa de diversos grupos que têm interesse nesta área. Apresentam-se a seguir algumas informações sobre os trabalhos mais representativos de alguns desses grupos mencionados nesse artigo e não tratados anteriormente.

2.3.8.1 - Projeto de Sistemas de Gerenciamento Baseado em Conhecimento e Problemas Orientado ao Tempo.

O grupo liderado por Gio Wiederhold tem interesse em duas áreas que envolvem questões relacionadas ao tempo :

- a) O reconhecimento de tempo em gerenciamento de BD e o relaxamento de restrições rígidas de tempo em sistemas distribuídos.
- b) O uso de BD que incorporem informações de tempo e como tal apresentem capacidade de representação e inferência que vão além daqueles em BD instantâneas convencionais.

2.3.8.2 - Estrutura Unificada Para Projeto e Gerenciamento de Bases de Dados Temporais.

Na Universidade da Flórida há um grupo envolvido com um projeto de BDT, liderado por S. Navathe.

Um modelo relacional temporal (TRM - "Temporal Relational Model") foi proposto visando incorporar semântica temporal do mundo real em BD. O modelo TRM suporta uma variação de atributo "time-stamp" dentro da estrutura clássica de BD relacionais para capturar o comportamento de atributos que variam com o tempo.

Os atributos que variam com o tempo são categorizados como atributos síncronos e assíncronos.

Uma linguagem de consulta que se propõe a ser amigável ao usuário, denominada TSQL foi proposta para recuperação de informações temporais. A linguagem TSQL é um super-conjunto da linguagem SQL, e possui novas construções sintáticas para manipular aspectos semânticos relacionados com tempo.

2.3.8.3 - O Projeto do SGBD POSTGRES.

As informações fornecidas foram extraídas do artigo [STON86] que apresenta o projeto preliminar de um SGBD relacional, denominado POSTGRES. A linguagem de consulta proposta para o POSTGRES é denominada POSTQUEL. Os aspectos a serem considerados neste trabalho, resumem-se apenas às informações relativas a tratamento de tempo apresentadas.

A linguagem POSTQUEL permite aos usuários salvar e consultar dados históricos.⁶ Os dados em uma relação não são eliminados ou atualizados. Recuperações convencionais sempre acessam as tuplas correntes na relação. Dados históricos podem ser acessados indicando-se o tempo desejado.

Aplicações que não desejam salvar os dados históricos podem especificar um ponto de corte ("cutoff") para a relação. Os dados que são anteriores ao ponto de corte são eliminados da base.

A linguagem POSTQUEL possui três níveis de hierarquia de memória : memória principal, memória secundária (disco magnético) e memória terciária (disco óptico). Dados correntes são armazenados em memória secundária e dados históricos migram para memória terciária. Contudo, esta distribuição dos dados armazenados não afeta o usuário.

Dentro da classificação adotada neste trabalho, a proposta de Stonebraker é classificada como base de dados "rollback".

2.4 - Como Embutir Relações Temporais em Relações Instantâneas.

Em virtude de várias das pesquisas em modelos de dados temporais terem-se focalizado em extensões para o modelo relacional, considera-se válido apresentar algumas maneiras pelas quais pode-se embutir relações temporais em relações instantâneas.

⁶A proposta do autor limita-se a tratar tempo físico.

Antes de abordá-las, faz-se menção à representação de tempo que pode ser feita por intervalos ou pontos de tempo, e também ao controle do registro de tempo que pode ser feito a nível de atributo ou objeto(tupla) "time-stamp". Esses assuntos são abordados no capítulo três.

A menos do trabalho de Segev e Shoshani [SEGE87], todas as outras pesquisas apresentadas neste capítulo utilizam o modelo relacional como modelo básico de BDT para embutir a relação temporal quadri-dimensional em uma relação instantânea bi-dimensional. Snodgrass apresenta em [SNOD87] algumas maneiras de se realizar esta tarefa.

A mais simples é adicionar dois atributos, cada um contendo um único valor de tempo (utiliza-se pontos de tempo). Clifford e Warren em [CLIF83] utilizam esta opção.

Outra é adicionar dois atributos, cada um contendo dois valores de tempo (utiliza-se intervalo de tempo). Ariav em [ARIA86] propôs esta representação, que também é utilizada em [SNOD87].

A terceira é incluir cinco atributos adicionais. Esta é semelhante a segunda, porém, com a adição do tempo em que a tupla é eliminada.

A quarta é associar valores de tempo com os próprios atributos. Dentro de uma tupla cada atributo pode ter diferentes valores em diferentes pontos do tempo. Nessa representação a relação não estará na Primeira Forma Normal (FNF). Gadia em [GADI88] utiliza esta opção.

2.5 - Considerações Finais.

Neste capítulo apresentou-se o que são bases de dados temporais, os conceitos que as permeiam, uma proposta de classificação e alguns trabalhos disponíveis na literatura que visam incorporar o tratamento de informações de tempo em modelos de dados e SGBDs.

Todos os trabalhos apresentados neste capítulo, exceto o proposto por Shoshani e Segev, utilizam o modelo relacional como modelo de dados básico. A partir de um modelo de dados, no caso o relacional, pode-se estender o modelo para desenvolver modelos de dados temporais, ou realizar o tratamento de tempo utilizando outra perspectiva.

Apesar do objetivo neste trabalho ser o tratamento de informações de tempo em um modelo de dados orientado a objetos, alguns dos aspectos discutidos referentes tanto ao modelo relacional como ao modelo proposto em [SEGE87] aplicam-se de um modo geral a qualquer modelo que pretenda tratar aspectos temporais, pois, dizem respeito às características semânticas das informações de tempo.

No próximo capítulo analisam-se algumas das propostas apresentadas neste capítulo frente a várias características que dizem respeito ao tratamento de informações de tempo em modelos de dados e SGBDs.

Capítulo 3

Características de Tempo

3.1 - Considerações Iniciais.

Segundo Ariav em [ARIA86], independente do modelo de dados utilizado as pesquisas na área de bases de dados temporais podem ser divididas em três perspectivas complementares, que são:

- a) formulação de uma semântica de tempo na modelagem conceitual de dados (tratamento de tempo em modelos de dados conceituais).
- b) o desenvolvimento de modelos de dados temporais que sejam semanticamente capazes de tratar aspectos temporais dos dados armazenados.
- c) o desenvolvimento de linguagens de consulta temporal.

O enquadramento de qualquer pesquisa fora de qualquer uma das três perspectivas é arbitrário, e baseado na percepção subjetiva de contribuições das respectivas pesquisas.

Ressalte-se que uma pesquisa pode englobar todas as três perspectivas, ou algumas delas, que são complementares. Por exemplo, Ariav em [ARIA86] desenvolve tanto um modelo (denominado TODM) para BDTs, como uma linguagem de consulta temporal (denominada TOSQL).

Essas perspectivas serão algumas das características utilizadas para analisar algumas propostas de tratamento de tempo em modelos de dados e SGBDs.

Neste trabalho, pretende-se enquadrar algumas propostas face às suas características marcantes, frente as perspectivas citadas anteriormente. Por exemplo, alguns autores afirmam desenvolver modelos para BDTs, mas, alguns deles fazem-no atribuindo a linguagem de consulta quase toda a tarefa para realizar o tratamento a aspectos temporais. Como exemplo, pode-se citar a proposta de Snodgrass em [SNOD87], em que é desenvolvido um modelo para BDTs, porém, as construções para manipular dados temporais são adicionadas a linguagem de consulta. Em consequência dessa abordagem, enquadra-se a proposta de Snodgrass somente na perspectiva de linguagem de consulta temporal.

Dado a ausência de um conjunto de características ou critérios estabelecidos e aceitos pela comunidade da área, pelos quais possa se analisar os modelos de dados temporais, decidiu-se definir um conjunto de características para fazê-lo neste trabalho.

Ariav cita em [ARIA86] que dada a ausência de linhas gerais para a avaliação de modelos de dados temporais, ele afirma que alguns modelos conceituais temporais parecem sugerir cinco maiores linhas gerais que são:

- tempo
- evento
- necessidade de ao menos duas dimensões de tempo (tempo válido e tempo da transação)

- o lugar do tempo na hierarquia das construções
- o critério para a determinação da existência de uma entidade

Avaliando-se as cinco linhas gerais, considera-se para efeitos de características neste trabalho, eventos e dimensões temporais. Tempo é o âmago deste trabalho.

A partir dos artigos pesquisados, selecionou-se as características de tempo a partir das quais serão analisadas algumas propostas de tratamento de tempo. Dentre as características presentes nos artigos pesquisados, algumas são conflitantes, outras dependentes de modelos específicos (como o relacional), outras consideradas como não necessárias por segmentos da comunidade, etc.

Na seção 3.12 apresenta-se uma tabela com um sumário da análise realizada.

Tratam-se a seguir as características de tempo selecionadas, com o enquadramento de algumas das propostas no tocante a cada uma delas.

3.2 - Tipo de Base de Dados em Relação à Classificação Adotada.

Algumas das primeiras pesquisas em BDT tratavam de tempo como sendo uma única dimensão. Clifford em [CLIF83] trata a dimensão de tempo válido. Lum e outros em [LUM*84] atentam para o fato de que o tempo possui duas dimensões que são tempo lógico e tempo físico, e que até a publicação do artigo [LUM*84] os autores não tinham notícia de nenhuma publicação que abordasse as duas dimensões de tempo. Snodgrass em [SNOD86a] além de tempo válido e tempo da transação, identifica uma outra dimensão que é o tempo definido pelo usuário. Ariav em [ARIA86] afirma também que ao menos duas dimensões de tempo (válido e transação) são necessárias em BDT. Vários artigos ([GADI88],[TANS89a], etc.) foram publicados paralelamente aos citados, em que se tratava apenas uma dimensão de tempo.

Decorrido algum tempo da publicação dos artigos de Lum, Ariav e Snodgrass, atualmente surgem indícios que tempo válido e tempo da transação estejam consolidados, e de que outras dimensões de tempo, tal como tempo definido pelo usuário, tenham sua existência reconhecida pela comunidade da área.

Como a classificação adotada neste trabalho baseia-se na capacidade dos modelos de suportarem os conceitos de tempo (válido e transação), apresenta-se a seguir a situação de algumas propostas, face a classificação.

Enquadra-se como base de dados "rollback", a proposta de [LUM*84], como base de dados histórica, as propostas de [CLIF83], [SEGE88], [GADI88], [TANS89a] e como base de dados temporal, as propostas de [ARIA86], [SNOD87].

3.3 - Modelo de Dados Básico.

A menos da proposta do modelo conceitual TDM em [SEGE87], todos as outras pesquisas analisadas propõem extensões ao modelo relacional para tratar tempo.

No caso de [SEGE87], os autores afirmam que o modelo TDM é independente de qualquer outro modelo, tais como, relacional e rede.

3.4 - Modelo Conceitual Temporal.

A questão aqui é se os modelos propostos formulam uma semântica de tempo em nível de modelo de dados conceitual, ou seja, o tratamento de tempo em modelos de dados conceituais.

Dentre os modelos pesquisados, o único a formular uma semântica de tempo em nível de modelo de dados conceitual é o TDM proposto por Segev e Shoshani em [SEGE87].

Snodgrass afirma, em [SNOD87], que modelos de dados que incorporem tempo diretamente são complexos e baseados em lógicas novas e não bem entendidas.

3.5 - Desenvolvimento de um Modelo Para Bases de Dados Temporais.

Nesta seção aborda-se o desenvolvimento de modelos de dados temporais que sejam semanticamente capazes de tratar aspectos temporais dos dados armazenados. Ou seja, que o modelo trate tempo diretamente.

As pesquisas nesta categoria tem-se focalizado na organização do armazenamento de dados temporais, e as operações necessárias para a sua manipulação significativa. O ponto de partida nestas propostas é exclusivamente o modelo relacional - examinam-se modificações e extensões para esse modelo necessárias para permitir o tratamento significativo de informações de tempo.

Para o desenvolvimento de um modelo para BDT, existem ao menos duas possibilidades:

a) estender a semântica do modelo de dados (relacional, p. ex.) para incorporar tempo diretamente.

b) utilizar os modelos de dados instantâneos (convencionais) com a informação sobre tempo aparecendo como atributo adicional. Neste caso, a lógica do modelo não incorpora tempo. Como essa abordagem concentra-se nas construções da linguagem de consulta úteis na manipulação de dados que contêm referências temporais, esta abordagem será enquadrada na seção 3.6, linguagem de consulta temporal.

Clifford e Warren em [CLIF83] escolheram a primeira opção, utilizando o modelo entidade-relacionamento para formular a lógica intencional (ILs). Esta lógica serve como um formalismo para a semântica temporal de BDTs. Também Ariav em [ARIA86], e Tansel em [TANS89a] utilizam a primeira opção.

3.6 - Linguagem de Consulta Temporal.

A ênfase nas pesquisas que baseiam-se principalmente em construções da linguagem de consulta para o desenvolvimento de modelos para dados temporais, é tipicamente no acesso do usuário aos dados. Ou seja, o modelo não incorpora tempo diretamente.

O modelo relacional instantâneo serve como o modelo básico de BDT. Cada relação temporal é embutida em uma relação instantânea contendo atributos temporais adicionais. Como a lógica do modelo não incorpora tempo diretamente, a linguagem de consulta deve

traduzir consultas e atualizações envolvendo tempo em recuperações e modificações nas relações instantâneas básicas. Gadia em [GADI88] e Snodgrass em [SNOD87] utilizam esta abordagem.

A linguagem TQuel ([SNOD87]) utiliza o modelo relacional como modelo básico para BDT. A linguagem TQuel não é baseada em um modelo de dados temporal explícito, e a ênfase é principalmente nos aspectos do compilador da linguagem de consulta temporal.

Ariav e Tansel propõem as linguagens de consulta temporal TOSQL [ARIA86] e TBE [TANS89a], porém, desenvolvem um modelo que trata tempo diretamente, tal como abordado na seção 3.5.

3.7 - Atributo/Objeto "Time-Stamp".

Na incorporação de tempo nos modelos de dados existem duas possibilidades para controlá-lo. Uma é objeto(tupla) "time-stamp" e a outra é atributo "time-stamp". Tansel em [TANS89a] afirma que tupla "time-stamp" cria redundância excessiva de dados e complexidade no modelo, além do que, tempo ajusta-se naturalmente a nível de atributo quando relações Não Primeira Forma Normal (NFNF) são permitidas. Em consequência, pode-se concluir que tempo ajusta-se naturalmente a nível de atributo em modelos orientados a objetos.

A opção de tupla "time-stamp" mantém as principais características da Primeira Forma Normal de bases de dados relacionais.

A abordagem atributo "time-stamp" não segue o modelo relacional padrão, mas, utiliza Não Primeira Forma Normal (NFNF), ou o "Nested Relational Model" (NRM) [TANS89b]. No NRM os valores dos atributos não requerem atomicidade, e podem ser formados por componentes compostos e multivalorados em uma lista arbitrária. O NRM é na realidade uma abordagem formalizada para representação de estruturas hierárquicas dentro de tuplas.

Na utilização de atributo "time-stamp" deve-se fazer algumas considerações que são:

- a) em geral, nem todos os atributos são temporais.
- b) apenas os atributos temporais possuem estrutura adicional para valores de tempo.
- c) para cada atualização do atributo, apenas a estrutura do atributo é afetada e não do objeto todo.

Na utilização de objeto "time-stamp" deve-se fazer algumas considerações que são:

- a) para cada atualização de qualquer atributo é necessária a criação de uma nova cópia de todo o objeto.
- b) não é necessário criar uma estrutura a nível de atributo para manipular valores de tempo.
- c) se existir mais de um atributo temporal na mesma relação, tem-se um único ponto (ou intervalo) de tempo associado com todos os valores dos atributos da tupla. Eventualmente isto conduz à definições de formas normais temporais [SEGE88].

Utiliza-se atributo "time-stamp" em [CLIF83], [ARIA86], [SEGE88], [GADI88], [TANS89a], e tupla "time-stamp" em [SNOD87] e [LUM*84].

3.8 - Ponto/Intervalo de Tempo.

Há dois tipos de dados temporais básicos. Dados do tipo evento ocorrem em pontos específicos do tempo, e são instantâneos (denominados pontos de tempo), enquanto que dados do tipo intervalo ocorrem em um intervalo de tempo.

De acordo com o tipo de ambiente alvo é conveniente utilizar uma ou outra representação. Em [SEGE88] os autores afirmam que pontos de tempo são convenientes para representar dados de eventos, tais como o registro do momento e a magnitude de terremotos, enquanto que intervalos de tempo são convenientes para dados com densidade passo constante. A propriedade densidade passo constante é definida na seção 2.3.2.2.

É importante observar que se o conceito de intervalo existe na linguagem de consulta, não é necessário representá-lo no modelo de dados temporal.

Tansel em [TANS89a] definiu o conceito de tripla, que possui dois valores que indicam o intervalo de tempo em que um terceiro valor (valor do atributo temporal) é válido.

Gadia em [GADI88] define o conceito de elemento temporal como sendo um conjunto de intervalos de tempo disjuntos.

O modelo desenvolvido por Snodgrass em [SNOD87] trata tanto ponto de tempo (evento) quanto intervalo de tempo. Para representar eventos existe um único valor de tempo. Para representar intervalos existem dois valores de tempo delimitando o intervalo.

3.9 - Propriedades Semânticas.

Propriedades semânticas de dados temporais, tais como granularidade, tempo de vida, densidade devem ser representadas nos modelos de dados que pretendam ser temporais.

Shoshani e Segev em [SEGE87] e [SEGE88] abordam explicitamente as propriedades semânticas acima mencionadas.

Os autores das pesquisas analisadas e apresentadas no capítulo dois abordam algumas das propriedades semânticas mencionadas, sem no entanto fazerem-no explicitamente.

No tocante a granularidade, assume-se que o sistema de base de dados temporal deve ter algum entendimento das unidades padrões de tempo utilizadas nas atividades humanas e seus inter-relacionamentos. Neste caso, o usuário deve escolher alguma granularidade mínima para cada aplicação, baseando-se na menor unidade de tempo que seja necessária. Assim, o usuário pode referenciar tanto a menor granularidade, quanto a unidades maiores de granularidade se necessário for, pois assume-se que o sistema reconhece o relacionamento entre as diferentes unidades de tempo. Deste modo, se a granularidade mínima escolhida for segundo, pode-se referenciar minutos, horas, que estes serão convertidos para a representação de segundos apropriada.

3.10 - Implementação.

A proposta de Snodgrass em [SNOD87] foi implementada, mas, parcialmente. O protótipo foi construído adicionado-se ao SGBD INGRES uma camada que realiza o

tratamento a aspectos temporais. Utilizou-se três pessoas-mês por um período de um ano para a construção do protótipo. Neste período não está incluída a familiarização com o SGBD INGRES e a linguagem Quel.

O SGBD POSTGRES ([STON86]) também foi implementado, porém, é classificado como SGBD "Rollback".

3.11 - Aplicação Alvo.

Shoshani e Segev em [SEGE87] afirmam que a motivação inicial para trabalharem com BDTs foram aplicações científicas e estatísticas.

Ariav em [ARIA86] afirma que sua motivação inicial foram sistemas de apoio a decisão.

3.12 - Considerações Finais.

Neste capítulo foram apresentadas características gerais pelas quais analisaram-se algumas das pesquisas em modelos de dados e SGBDs apresentadas no capítulo dois.

No próximo capítulo apresenta-se o **Modelo de Representação de Objetos - MRO***.

Observação : Na tabela 3.1 as linhas representam as características e as colunas os autores e os artigos.

- (1) Tipo de BD.
 - (2) Modelo de dados básico.
 - (3) Modelo conceitual temporal.
 - (4) Desenvolvimento de um modelo para BDTs.
 - (5) Linguagem de consulta temporal.
 - (6) Atributo/objeto "time-stamp".
 - (7) Ponto/intervalo de tempo.
 - (8) Propriedades semânticas.
 - 8.1 Granularidade
 - 8.2 Tempo de vida
 - 8.3 Densidade
 - (9) Implementação.
 - (10) Aplicação alvo.
-
- (A) Segev e Shoshani ([SEGE87],[SEGE88]).
 - (B) Snodgrass ([SNOD87]).
 - (C) Tansel e outros ([TANS89a]).
 - (D) Ariav ([ARIA86]).
 - (E) Clifford e Warren ([CLIF83]).
 - (F) Gadia ([GADI88]).

Tabela 3.1 - Sumário da análise dos modelos

	A	B	C	D	E	F
1	histórica	temporal	histórica	temporal	histórica	histórica
2	TDM e relacional	relacional	relacional	relacional	relacional	relacional
3	sim	não	não	não	não	não
4	sim	não	sim	sim	sim	não
5	sim	sim	sim	sim	sim	sim
6	atributo tupla	tupla	atributo	atributo	atributo	atributo
7	ponto	ponto e intervalo	intervalo	intervalo	ponto	intervalo
8.1	sim	sim	sim	sim	sim	sim
8.2	sim	sim	sim	sim	não	sim
8.3	sim	não	não	não	não	não
9	não	sim	não	não	não	não
10	científicas e estatísticas	?	?	sistemas de apoio a decisão	?	?

? - Não especificado no(s) artigo(s).

Capítulo 4

O Modelo de Representação de Objetos

4.1 - Considerações Iniciais.

O **Modelo de Representação de Objetos - MRO*** [TRAI86][TRAI88] tem sido elaborado no Departamento de Ciências de Computação e Estatística do ICMSC-USP em São Carlos desde o final de 1986, com o objetivo de permitir a construção de Sistemas de Gerenciamento de Bases de Dados que atendam às necessidades de manipulação de dados e representação de informações dos sistemas de apoio a projetos de engenharia ("Computer Aided Design" - CAD), ou de ambientes de suporte ao desenvolvimento de "softwares" ("Computer Aided Software Engineering" - CASE).

Este capítulo tem por objetivo introduzir os principais conceitos que envolvem o **MRO***, bem como os subsistemas que compõem o **GERenciador de Objetos - GEO**, um protótipo de **SGBDOO** apoiado no **MRO***.

Para efeito de apresentação do Modelo de Representação de Objetos, são descritos sucintamente seus conceitos e definições. A descrição completa de todos os conceitos, bem como o formalismo matemático definido para o **MRO***, podem ser encontrados em [BLAJ92].

4.2 - Conceituação.

Nesta seção apresentam-se conceitos inerentes ao **MRO***.

4.2.1 - Esquema de Dados Utilizado nos Exemplos.

Com o objetivo de tornar claro os conceitos apresentados neste capítulo e nos próximos, será utilizado um esquema de uma aplicação, do qual são retirados os exemplos apresentados. A aplicação consiste de um sistema de informações financeiras que armazena informações de um Banco. Exemplos de informações financeiras são depósitos a vista, depósitos a prazo, poupança, captação de CDB/RDB, etc. Esta aplicação apresenta o desempenho de unidades organizacionais (agências, departamentos, diretorias, etc) de uma instituição financeira, em função de indicadores econômicos (cruzeiro, dólar, marco alemão, libra esterlina, etc). O Esquema e os dados dos exemplos encontram-se no apêndice A.

4.2.2 - Diagramas de Representação.

No MRO* a representação gráfica de seus dados é realizada através dos seguintes diagramas:

Figura 4.1 - DRO da Aplicação de Informações Financeiras

a) **Diagrama de Representação de Instâncias - DRI** : onde pode-se representar os dados que estão armazenados na Base de Dados. Sua função é apresentar apenas as informações de interesse de um usuário em um determinado momento.

b) **Diagrama de Representação de Objetos - DRO** : onde são representados os Tipos, Relacionamentos e Atributos que compõem o Esquema de Dados. Nele pode-se representar o Esquema de Dados completo ou apenas uma parte em que se demonstre interesse.

c) **Diagrama Hierárquico de Colônias - DHC** : onde se representa a Hierarquia de colônias em que se está operando.

As figuras 4.1 e 4.2 apresentam respectivamente um DRO e um DRI da aplicação de informações financeiras.

4.2.3 - Fundamentos.

O MRO* modela as entidades do mundo real como **Objetos**, os quais podem associar-se a outros Objetos através de **Relacionamentos**. Tanto Objetos quanto Relacionamentos são caracterizados por **Atributos**. Cada atributo possui um tipo e um conjunto não vazio de valores. Objetos semelhantes são caracterizados através de **tipos de objetos**. Todo Objeto possui um **Código** gerado pelo sistema, que o identifica univocamente durante toda a sua existência na Base de Dados. Além disso, um Objeto possui identificadores ou sinônimos fornecidos pelos usuários.

O significado de um Atributo pode ser reconhecido pelo SGBD através de sua **Característica** (sinônimo, comentário, tempo, estrutura, gráfico, imagem, regra, visualização, propriedade, procedimento) e seu **Tipo de Dado** (byte, int, long, coditipo, float, double, cadeia, 2dcoordenadas, data, hora, valor).

A forma mais simples de relacionamento é chamada **Relacionamento Binário**, ou simplesmente **relacionamento**, e associa dois Objetos denominados **objeto origem** e **objeto destino**, através de **Modalidades**. Todo relacionamento segue um padrão definido no esquema da aplicação denominado de **tipo de relacionamento binário**, onde define-se o tipo do objeto origem, tipo do objeto destino e a modalidade permitidos para esse relacionamento. Cada relacionamento binário existente possui um **relacionamento oposto**, para representar a sua semântica oposta. Neste relacionamento os objetos origem e destino têm seus papéis trocados. Uma associação mais elaborada é chamada **Relacionamento Triplo**, sendo composta por dois ou mais Relacionamentos Binários que possuem em comum um objeto origem. Todo relacionamento triplo segue um padrão definido no esquema da aplicação denominado de **tipo de relacionamento triplo**, onde são especificados a modalidade de relacionamento e os tipos de objetos envolvidos.

Para tornar a modelagem mais natural, são mantidos na Base de Dados certos Relacionamentos e Atributos que não precisam ser definidos explicitamente pelo usuário. Tais elementos são ditos **Intrínsecos**. Um exemplo é o atributo que armazena uma informação indicando que usuário criou um objeto na Base.

Exemplificando os conceitos citados anteriormente, apresenta-se um Diagrama de Representação de Objetos - DRO e um Diagrama de Representação de Instâncias - DRI nas figuras 4.1 e 4.2, respectivamente. Na figura 4.1 os Objetos são representados por círculos (Indicador, Informação Financeira, Unidade Organizacional, Aplicação), Relacionamentos Binários representados por arcos direcionados e usando suas respectivas Modalidades (Possui, Expressa), e Atributos representados por traços ligados aos Objetos ou Relacionamentos correspondentes (Nome, Tipo, Classificação, Sigla, Valorização Indicador, etc...). Na figura 4.2 os Objetos são representados por círculos (Informação_Financeira Depósito_a_Vista, Uniorg São_Carlos, Indicador Dólar), Relacionamentos Binários representados por arcos direcionados e usando suas respectivas Modalidades (Possui, Expressa) e Atributos representados por traços ligados aos Objetos ou Relacionamentos correspondentes (São Carlos, Agência, Depósito_a_Vista, Consolidado, Captação, etc...).

Os objetos da base são divididos em partições, denominadas **Colônias de Objetos**, ou

Figura 4.2 - DRI da Aplicação de Informações Financeiras

simplesmente, **Colônias**. Cada colônia é constituída por um conjunto de objetos, e se refere a um determinado aspecto de interesse. Os objetos que pertencem a uma colônia são chamados **habitantes** dessa colônia. Cada objeto habita exatamente uma colônia. As colônias são classificadas em **Tipos de Colônias**. Cada tipo de colônia determina um conjunto de tipos de objetos. Assim objetos de um dado tipo podem somente habitar colônias de um único tipo.

Os objetos que habitam uma colônia tornam-se disponíveis quando a colônia é aberta para acesso. Das várias instâncias de colônias de cada tipo que podem existir apenas uma pode estar aberta num dado instante. Uma das colônias deve estar permanentemente disponível e é denominada **Colônia Global**. Os objetos da colônia global são os que estão disponíveis quando se está no nível menos detalhado de acesso ao sistema e permanecem sempre disponíveis. Existe também um **Tipo de Colônia Global** que contém os tipos de objetos existentes na colônia global.

O conceito de colônia é distinto do conceito de classes de objetos: enquanto uma classe corresponde a todos os objetos de um mesmo tipo uma colônia inclui objetos de diversos tipos diferentes e - a menos da colônia global não exaure o conjunto dos objetos de cada

um dos tipos que a habitam.

Os tipos de colônias organizam-se segundo uma hierarquia na qual a colônia global é o topo da hierarquia, o nível imediatamente subordinado é constituído pelas **colônias constringidas** por objetos de tipos que habitam a colônia global e assim sucessivamente. Essa hierarquia ocorre devido à existência de um relacionamento intrínseco de composição entre o objeto que constringe a colônia e os objetos que a habitam.

O conceito de colônias suporta o controle de **versões e alternativas** de um projeto permitindo a geração de **instâncias de colônia de objetos**. Uma colônia pode ter várias instâncias sendo cada uma chamada de **variante**. Podem existir variantes que permitem alterações no conteúdo da colônia - que correspondem às **alternativas** de projeto ainda em desenvolvimento - ou variantes já congeladas, denominadas **versões**, que não podem mais ser alteradas.

Os tipos de objetos e de relacionamentos definidos no esquema podem ser especializados em **Subtipos de objetos e Subtipos de relacionamentos**. Subtipos podem ser novamente refinados em novos subtipos ou reagrupados com outros tipos e subtipos formando outros subtipos. Subtipos são mantidos através de relacionamentos de generalização intrínsecos. Um Subtipo é especificado através da indicação de:

a) um tipo, ou um conjunto de subtipos já previamente especificado, que será denominado Tipo Mestre do Subtipo;

b) um predicado, que estabelece o critério de como escolher os objetos do Tipo Mestre que constituirão o conjunto de objetos daquele subtipo, através da indicação de atributos e relacionamentos;

c) um conjunto padrão de valores de atributos e de relacionamentos que podem ser usados para inicializar os valores ainda não definidos de um objeto desse subtipo quando este é criado.

4.2.4 - Esquema e Meta-Esquema de Dados.

Considerando que cada Colônia, Objeto, Relacionamento e Atributo possui seu respectivo Tipo, pode-se descrever um esquema genérico do modelo MRO^{*} através de um **Meta-Esquema de Dados** (figura 4.3), onde cada elemento pode ser encarado como um Meta-Tipo, ou seja, uma definição do que é "ser um Tipo" e qual a sua relação com os outros Tipos.

A existência de um Meta-Esquema de Dados permite então, definir o **Esquema de Dados** de uma aplicação como um conjunto de informações representadas em uma estrutura sintaticamente idêntica à usada na representação dos dados da Base, sendo, portanto o Esquema, uma Base de Dados (Meta-Base de Dados) com a mesma estrutura da Base de Dados que ela define e assim, o Esquema pode ser armazenado em uma Colônia de Objetos, denominada **Colônia Esquema**.

4.3 - GERenciador de Objetos - GEO.

Em meados de 1988, iniciou-se o desenvolvimento de um SGBDOO denominado GERenciador de Objetos [TRAI91] que incorpora os conceitos do MRO*. Um dos objetivos desse sistema é a validação prática e o estudo de como os conceitos do MRO* e de Sistemas Orientados a Objetos em geral, podem ser implantados em ferramentas reais.

Para a tarefa de implementar o Gerenciador de forma que este possa manipular eficientemente os dados o GEO foi dividido em 5 módulos básicos :

1.) Núcleo de Acesso a Registros: emula um sistema de Gerenciamento de Arquivos Lógicos, fornecendo suporte aos demais módulos. É composto pelo Subsistema de Gerenciamento de Registros (SGR), Subsistema de Gerenciamento de Arquivos (SGA) e Subsistema de Gerenciamento de Memória (SGM). Através destes módulos é realizado todo o gerenciamento de memória secundária, simulação de uma memória "cache" para disco em memória e controle de acesso e proteção dos dados, através do gerenciador de transações.

2.) Subsistema Básico de Gerenciamento de Objetos: possibilita a manipulação de Objetos sem restrições de ordem semântica. Envolve o tratamento de Identificadores de Objetos, manipulação de Composições de Objetos, contexto e navegação na Base.

3.) Subsistema Básico de Gerenciamento de Relacionamentos: permite a manipulação de Relacionamentos (binários ou triplos) definidos pelo usuário. O GEO gerencia Relacionamentos Intrínsecos e Relacionamentos Extrínsecos - os intrínsecos são aqueles cuja semântica é conhecida pelo Gerenciador, os extrínsecos são aqueles cuja semântica é definida pelo usuário e são gerenciados por este módulo em questão.

4.) Subsistema Básico de Gerenciamento de Atributos: permite a manipulação de Atributos com características de Propriedades e Comentários, tanto para Objetos quanto para Relacionamentos.

5.) Gerenciador de Esquemas de Dados: consiste nas Primitivas de Acesso ao Esquema, as quais são ativadas pelos demais módulos.

Apenas os Subsistemas Básicos de Gerenciamento estão completamente implementados. Assim as operações devem ser feitas sem levar em consideração os acessos mais elaborados ao MRO*, tal como acesso automático a Subtipos de Objetos e de Relacionamentos. Correspondendo a cada Subsistema de Gerenciamento Básico está sendo realizada a implementação de um Subsistema de Gerenciamento Semântico. O conjunto de Subsistemas de Gerenciamento Básico é denominado GEO/MRO-Básico e o conjunto de Subsistemas de Gerenciamento Semântico é denominado GEO/MRO-Semântico.

O GEO/MRO-Básico efetua todas as operações de acesso efetivo na Base de Dados, tal como criar ou eliminar Objetos, Relacionamentos, etc. Já o GEO/MRO-Semântico deverá realizar as mesmas operações, porém avaliando não apenas as regras sintáticas estáticas como é feito agora pelo GEO/MRO-Básico, mas também as regras semânticas que dependem do ambiente instantâneo dos dados na Base de Dados. A estrutura geral do

Gerenciador é mostrada na figura 4.4.

Aplicativos			
Editor de Esquemas	Editor de Consultas	Aplicativos Específicos	...
MRO-Semântico			
MRO-Básico			
Núcleo			
Sub-Sistema de Gerenciamento de Registros			
Registros Fixos		Registros Variáveis	
Sub-Sistema de Gerenciamento de Arquivos			
Registros Fixos		Registros Variáveis	
Sub-Sistema de Gerenciamento de Memória			
Sistema Operacional			

Figura 4.4 - Estrutura do GEO

4.4 - Considerações Finais.

Neste capítulo descreveu-se o MRO* e a implementação de um gerenciador nele apoiado (GEO). O desenvolvimento deste modelo tem como objetivo fundamental suportar as necessidades de aplicações não convencionais de grupos de pesquisa, principalmente envolvendo seus projetos, permitindo que gerenciadores apoiados nesse modelo apresentem considerável eficiência e completeza.

No próximo capítulo realizam-se considerações sobre o MRO* e o tempo, levando-se em consideração as características apresentadas no capítulo três.

Capítulo 5

Conceitos Temporais No MRO*

5.1 - Considerações Iniciais.

Em aplicações de suporte a projetos apoiadas em gerenciadores de BD pode-se considerar que o produto final de uma atividade de se projetar algo é a própria base de dados. Assim ela deve ser capaz de representar tão precisamente quanto possível todos os aspectos levados em consideração nas atividades de projeto, incluindo-se aí os aspectos estruturais, comportamentais e semânticos. Dentre os últimos, constituem-se como fundamentais aqueles que representam a evolução progressiva de um projeto, e o acompanhamento da execução de um projeto no mundo real. Esses conceitos envolvem principalmente o tempo e controle de versões.

Versões podem ser encaradas como visões completas e estáticas de um objeto (o projeto) em evolução, tal como instantâneos tirados ao longo do tempo desse objeto. Os requisitos levados em consideração para se definir uma versão são lógicos, como por exemplo o conjunto de regras que se leva em consideração para determinar quando uma versão está completa.

Já o tempo é uma representação de uma sequência de eventos que se sucede de uma maneira linear - que pode ser quantificado através de uma unidade (absoluta ou relativa) e dessa forma pode ser encarado como uma maneira física de se representar a evolução de um projeto - dessa maneira podem ser consideradas ortogonais entre si.

Por outro lado, versões são construídas através de operações que podem ser sequencializadas e portanto instanciadas em alguma escala de tempo. Nesse sentido, a representação e o gerenciamento do tempo podem ser considerados como um refinamento da representação e gerenciamento de versões. Assim pode-se encarar que diferentes versões podem ser obtidas através da evolução temporal das informações armazenadas em uma base de dados.

Para este trabalho, considera-se que um gerenciador de BD para apoio a aplicações de suporte a projeto deve contar com recursos de representação e gerenciamento de tempo e de versões. Seu escopo concentra-se no tratamento de informações temporais, que podem ser quantizadas e sequencializadas de uma maneira física. Não é tratado o aspecto lógico da evolução dos dados armazenados, e portanto o conceito de versão não é tratado. No entanto isso foi levado em consideração no desenvolvimento do trabalho sob dois pontos de vista: primeiro, restringindo-se apenas à aspectos temporais de fato e não tentando dar suporte à necessidades de evolução de um projeto, pois considerou-se que trabalhos na linha de gerenciamento de versões devem complementar este trabalho; segundo, buscando não colocar conceitos que possam vir a dificultar um futuro suporte integrado ao controle de versões criando-se construções temporais que possam vir a conflitar com estas.

Assim, desenvolveu-se um modelo conceitual voltado exclusivamente ao suporte de

tempo tal como verificou-se necessário para algumas aplicações (dentre elas as de projeto), estendendo-se o MRO^{*} nesse sentido, bem como especificaram-se as operações que manipulam dados com aspectos temporais e desenvolveram-se as primitivas correspondentes no GEO.

Para a definição de um modelo de dados, dois aspectos devem ser tratados. O primeiro aspecto é a especificação das **construções de dados** do modelo. O segundo aspecto é a especificação de **operações válidas** nas construções de dados do modelo. Essas operações facilitam o acesso aos dados e a manipulação do conteúdo da BD. Os aspectos semânticos foram considerados na concepção desse modelo.

Neste capítulo mostra-se a extensão feita no MRO^{*} para suportar informações temporais. Como os conceitos já foram genericamente discutidos no capítulo dois, este capítulo restringe-se à descrição de como eles foram incorporados ao MRO^{*}.

5.2 - Conceitos Temporais.

5.2.1 - Dimensões Temporais.

Em BD considera-se que o tempo é composto por várias dimensões temporais [SNOD86a] linearmente independentes. Segundo a classificação de BDT proposta por Snodgrass em [SNOD86a], ao menos as dimensões tempo válido e tempo da transação devem ser representadas para que uma BD possa ser denominada temporal. Outra dimensão temporal denominada tempo definido pelo usuário também pode ser representada mas não é um critério para a classificação proposta por Snodgrass.

Deve-se ressaltar que as dimensões temporais tempo da transação e tempo válido são interpretadas (tratadas) pelo SGBD temporal e que a dimensão tempo definido pelo usuário têm sua semântica definida e interpretada por funções definidas pelo usuário e não pelo SGBD. Para a dimensão tempo definido pelo usuário apenas a representação interna e funções de entrada e saída são necessárias.

Neste trabalho mostra-se como foi incorporado ao MRO^{*} a capacidade de representar as dimensões temporais⁷ tempo válido, tempo da transação e tempo definido pelo usuário⁸.

A representação de mais do que uma dimensão temporal no mesmo atributo afetando a mesma unidade de informação na base é uma tarefa complexa pois elas devem ser compatibilizadas para a realização de operações na BD e influenciam na definição do armazenamento das informações afetadas pelo tempo. Em determinados casos particulares a representação de mais do que uma dimensão temporal num mesmo atributo é

⁷Neste trabalho os termos dimensão temporal e atributo temporal referem-se a um mesmo conceito: o fato de utilizar-se um atributo da base para representar um valor do tempo segundo uma perspectiva (dimensão).

⁸Pode-se definir mais que uma dimensão temporal do tipo tempo definido pelo usuário.

conceitualmente significativo e por isso decidiu-se pelo suporte a isso na BD. No entanto, a avaliação do significado genérico da associação de dois ou mais atributos temporais num mesmo atributo temporizado não foi empreendida. Este trabalho procura resolver o problema da representação e do suporte computacional desse fato, mas não é feita nenhuma análise conceitual do significado que isso implica.

5.2.2 - Atributo/Objeto Temporizado.

Para a incorporação de tempo nos modelos de dados existem duas possibilidades. Uma é rotular objetos com "time-stamp" (objeto temporizado) e a outra é rotular atributos com "time-stamp" (atributo temporizado).

A abordagem atributo temporizado⁹ envolve cada valor de cada atributo temporizado sendo rotulado com um valor de tempo, enquanto que na abordagem objeto temporizado um objeto é rotulado com um valor de tempo.

Valores de tempo para atributos e objetos são compostos por valores de tempo para cada uma das dimensões temporais associadas ao atributo ou objeto.

A abordagem objeto temporizado requer menos esforço de implementação mas uma grande quantidade de operações de manipulação pelo SGBD para acessar informação temporal. Atributo temporizado requer mais esforço de implementação mas a recuperação de informação temporal é feita mais facilmente (mas não necessariamente mais rápida) e utilizando um volume de dados armazenado menor.

A abordagem objeto temporizado cria redundância excessiva de dados pois quando o valor de um atributo temporizado muda todo o objeto tem que ser duplicado, com excessão do atributo causador da alteração. Considerando o grande volume de dados temporais esta redundância agrava o problema do armazenamento e recuperação de dados em um SGBD temporal. No caso de atributo temporizado os dados temporais são modelados e armazenados associados ao objeto a que pertencem não ocorrendo a duplicação do objeto.

A abordagem atributo temporizado é mais próxima da visão do usuário pois para um determinado objeto existe apenas uma versão armazenada, com os dados temporais armazenados nos atributos, compondo o objeto de acordo com a especificação das restrições dos atributos e dos valores de tempo especificados. No caso de objeto temporizado, para um determinado objeto têm-se várias versões do mesmo objeto armazenadas devido ao tempo associado, que não é a visão que o usuário tem de um único objeto.

Não há informações a respeito de resultados de implementação de BDTs que utilizem especificamente a abordagem atributo temporizado. A figura 5.1 ilustra o atributo sofrendo a ação do tempo, que por sua vez é composto por dimensões temporais. Neste trabalho optou-se pela utilização da abordagem atributo temporizado na incorporação de tempo ao MRO*.

⁹Atributo temporizado é um atributo qualquer que esteja associado a um ou mais atributos temporais. Atributo temporal é aquele que guarda a informação de uma dimensão temporal.

5.2.3 - Intervalo/Ponto de Tempo.

Para a representação de tempo em BD existem ao menos duas opções que são ponto e intervalo de tempo.

A utilização de ponto de tempo requer menos espaço de armazenamento do que intervalo de tempo, porém, necessita de maior capacidade de representação para manipulação de dados temporais. Por outro lado, utilizando-se intervalo de tempo, as informações sobre limite inferior e superior associadas a uma instância de um atributo estão disponíveis "junto" à instância, enquanto que utilizando-se ponto de tempo o limite superior não é armazenado junto à instância do atributo, o que causa ao menos um outro acesso ao

dispositivo de armazenamento para se obter o limite superior ou informação equivalente.

Em aplicações em que a propriedade densidade é passo constante, o limite superior associado a uma instância de um atributo é o mesmo que o limite inferior do seu sucessor, ou seja, da instância seguinte do atributo, caso exista. Para tais aplicações é possível utilizar a opção de ponto de tempo sem a preocupação de como obter o limite superior associado a uma instância do atributo.

Figura 5.1 - Dimensões temporais agindo em um atributo

5.2.4 - Semântica Temporal.

Propriedades semânticas de dados temporais tais como granularidade e tempo de vida devem ser representadas nos modelos de dados temporais. As propriedades semânticas representadas no MRO^{*} são apresentadas na seção 5.3.4.

5.3 - Como os Conceitos Temporais Afetam o MRO^{*}.

A seguir apresentam-se considerações a respeito da incorporação de conceitos de dados temporais - tais como dimensões temporais e propriedades semânticas - ao MRO^{*}.

5.3.1 - Atributos Temporais.

O MRO^{*} suporta as dimensões temporais tempo válido, tempo da transação e tempo definido pelo usuário. Quando um atributo é temporizado esta situação é representada no esquema por:

<atributo_temporizado> (Θ atributo_temporal (<atributo_temporal>)*)¹

e na base por:

<atributo_temporizado> <-- valor (Θ atributo_temporal <-- limite_inferior, limite_superior) (atributo_temporal <-- limite_inferior, limite_superior)*

Caso o limite superior não seja informado, têm-se duas opções. Representar somente o limite inferior recebendo um valor de tempo, ou o limite inferior recebendo um valor de tempo e o limite superior recebendo o valor ∞ "infinito".

Figura 5.2 - DRO com definição de atributos temporizados

Um exemplo de um DRO com definição de atributos temporizados é apresentado na figura 5.2. Um exemplo de um DRI com definição de atributos temporizados é apresentado na figura 5.3.

As abreviações utilizadas para representar os atributos temporais são:

tv : tempo válido

tt : tempo da transação

tu_i : tempo *i* definido pelo usuário

¹O símbolo de um relógio Θ é utilizado para representar tempo na definição de atributos temporizados.

5.3.2 - Atributo Temporizado.

No MRO^{*} decidiu-se controlar tempo através de atributo, ou seja, a abordagem utilizada é atributo temporizado. Existem dois benefícios nesta abordagem. O primeiro consiste em armazenar dados em um espaço menor do que o necessário para objeto temporizado, pois apenas os atributos de um objeto que sejam dependentes de tempo precisam ser representados várias vezes. O segundo benefício é que um atributo pode ter seu valor definido em função de tantos atributos temporais quanto for necessário, ao mesmo tempo em que, mesmo num único objeto se um atributo depende de determinado conjunto de atributos temporais e outro atributo depende de outro conjunto, cada atributo pode ser definido em função dos atributos temporais que realmente o afetam, independente dos atributos temporais que governem outros atributos temporizados do mesmo objeto, ou de outros objetos. Para exemplificar, suponha dois atributos: a_1 temporizado por tempo da transação e tempo 1 definido pelo usuário e a_2 temporizado por tempo válido e tempo 2 definido pelo usuário. Caso a opção fosse objeto temporizado, todos os atributos de um objeto que utilize os atributos a_1 e a_2 estariam temporizados pelos atributos temporais tempo da transação, tempo válido e tempos 1 e 2 definidos pelo usuário.

Figura 5.3 - DRI com definição de atributos temporizados

A partir da opção por controlar tempo a nível de atributo, alguns componentes do MRO tornam-se claramente passíveis de tratamento temporal. São objetos, relacionamentos e relacionamentos triplos, pois, esses componentes podem ter atributos

associados, sejam eles temporais ou não. Não é feita a associação de atributo à colônia e a incorporação de atributos a colônias tem consequências que não foram estudadas e fogem ao escopo deste trabalho. Consequentemente, colônias não serão passíveis de tratamento temporal.

A figura 5.4 ilustra os componentes do MRO^{*} passíveis de tratamento temporal.

5.3.3 - Intervalo de Tempo.

A abordagem intervalo de tempo foi adotada neste trabalho porque requer menos esforço de implementação do que o necessário para ponto de tempo, e além disso permite a representação de pontos de tempo, ao passo que a representação de pontos de tempo requer um esforço considerável para permitir a representação de intervalos. Na representação utilizada, armazena-se o valor de um atributo temporizado e os limites inferior e superior do intervalo de tempo referente ao valor do atributo. No caso de eventos que ocorrem em um instante t eles são representados por $[t, t + 1]$. Quando o valor do limite superior não for especificado, convencionou-se que o limite superior é ∞ (infinito). O limite superior quando "infinito" é limitado pela propriedade tempo de vida associada ao objeto. Em aplicações em que a propriedade tipo é passo constante, o limite superior associado a uma instância de um atributo é o mesmo que o limite inferior do seu sucessor, caso exista.

5.3.4 - Propriedades Semânticas.

Nesta seção apresenta-se a representação das propriedades semânticas no MRO^{*}.

5.3.4.1 - Granularidade.

No MRO^{*} a granularidade está associada ao relacionamento entre o atributo temporal e propriedade, propiciando desta maneira mais flexibilidade ao usuário. A granularidade dos atributos temporais dos atributos temporizados de um objeto devem ser compatíveis, i. é, todas calendário ou ordinal.

No MRO^{*} as propriedades semânticas granularidade, tipo e se a informação temporal é do tipo evento ou intervalo, são atributos do relacionamento entre tempo e atributo com característica de propriedade. A figura 5.5 do meta-esquema do MRO^{*} Temporal ilustra o conceito descrito.

Figura 5.4 - Componentes do MRO afetados

5.3.4.2 - Tempo de Vida.

Figura 5.5 - Meta-Eschema do MRO* Temporal

A propriedade tempo de vida está associada a cada componente do MRO* que seja passível de tratamento temporal, no caso objetos e relacionamentos, e determina o relacionamento "tempo de vida do" com objeto e relacionamento (binário e triplo). A figura

5.5 do meta-esquema do MRO* Temporal ilustra o conceito descrito. O tempo de vida associado a cada componente é visto como uma propriedade temporal do componente considerando a sua existência e necessita ser expresso somente uma única vez.

No tempo de vida devem estar todas e apenas as dimensões que temporizam os atributos, sejam eles associados a objetos ou relacionamentos. Existe um componente do tempo de vida para cada um dos atributos temporais representados, com os respectivos limites inferior e superior.

A seguir apresenta-se o significado do tempo de vida associado a cada atributo temporal:

- Tempo Válido: define a extensão dos pontos de tempo válidos para o atributo tempo válido.

- Tempo da Transação: o tempo de vida inicial é o momento em que o objeto é armazenado na base. O tempo de vida final indica o momento em que o objeto foi eliminado, caso isto ocorra.

- Tempo Definido Pelo Usuário: indica a extensão dos pontos de tempo em que o tempo definido pelo usuário é aplicável, de acordo com a definição especificada pelo usuário.

Ao menos quatro alternativas são possíveis para a atribuição de valores aos componentes do tempo de vida:

- a) Obrigar o usuário a definir explicitamente.
- b) Assumir o que está definido para o Ambiente
- c) Para cada tipo de objeto poder-se-á definir como tratar o tempo de vida na BD.
- d) Só assumir automaticamente a adoção de valores para o tempo da transação que compõe o tempo de vida; a adoção de valores para os outros atributos temporais fica para o usuário; o usuário pode atribuir valores posteriormente para cada um deles. Esta foi a opção adotada.

Quando um objeto ou relacionamento é temporizado o tempo de vida é associado automaticamente a ele. Os valores do tempo de vida podem ser redefinidos pelo usuário a qualquer instante. Os valores que estão no tempo de vida "atuam" como limitantes para os valores de tempo de cada atributo temporal associados a atributos temporizados. Sempre que forem modificados valores nos atributos temporizados, os limites de tempo serão os "regidos" pelo tempo de vida. Em cada acesso aos atributos temporizados o tempo de vida atua como um filtro restringindo o universo de pesquisa.

As operações desenvolvidas para a manipulação do tempo de vida são DEFINE e SUBSTITUIÇÃO e são apresentadas no capítulo seis (LAMRO).

5.3.4.3 - Densidade.

A propriedade densidade do tempo é representada no MRO*, somente quando for contínua ou discreta. Quando ela for **contínua** a função de interpolação necessária para definir valores não armazenados não é proporcionada.

5.3.5 - Considerações Sobre Propriedades Semânticas no MRO*.

A figura 5.6 ilustra o relacionamento entre as propriedades semânticas e os atributos

temporais, e é utilizada em aplicações, devendo ser utilizada na definição das mesmas. Para cada atributo temporal que uma aplicação utilizar a coluna correspondente deve ser preenchida de acordo. As seguintes observações aplicam-se a figura 5.6:

	SGBD trata automaticamente		SGBD não trata automaticamente			
	tempo da transação	tempo válido	tempo 1 def. p/o usuário	tempo 2 def. p/o usuário		tempo n def. p/o usuário
granularidade						
intervalo /evento						
densidade						
tempo de vida						

Figura 5.6 - Relacionamento entre props. semânticas e atributos temporais

a) O SGBD temporal deve tratar implicitamente os atributos temporais tempo da transação e tempo válido. O SGBD temporal também deve tratar o atributo temporal tempo definido pelo usuário, porém seguindo a resenha determinada pelo usuário.

b) A granularidade afeta o relacionamento entre o atributo temporizado e os atributos temporais aos quais o atributo esteja associado.

c) O tempo de vida associado a cada objeto ou relacionamento delimita os valores do objeto e do relacionamento em cada um dos atributos temporais que estejam associados a ele.

d) A propriedade densidade afeta o relacionamento entre o atributo temporizado e os atributos temporais aos quais o atributo esteja associado. Cada uma das possibilidades da propriedade densidade (passo constante, discreta, contínua) determina como são interpretados os valores de tempo de cada atributo temporal. Por exemplo, para o atributo temporal tempo válido associado a um atributo cuja propriedade densidade é passo constante, interpreta-se que uma instância do atributo é válida desde o limite inferior até o limite superior do intervalo de tempo associado ao atributo.

e) A granularidade definida para os atributos temporais que temporizam os atributos de um objeto ou relacionamento deve ser compatível, ou seja, ordinal ou calendário, possuindo a mesma granularidade.

f) Os atributos temporais que temporizam os atributos de um objeto ou relacionamento devem ser todos definidos ou como intervalo, ou como evento.

g) Os atributos temporais que temporizam os atributos de um objeto ou relacionamento devem ser compatíveis quanto a propriedade densidade; são compatíveis as densidades passo constante e contínua; a densidade discreta é incompatível com os outros dois.

5.3.6 - Definição de Atributo Temporal no Esquema.

As seguintes considerações referem-se à figura 5.5 do meta-esquema do MRO^{*} Temporal.

Na figura 5.5 o tempo é representado como as outras características gerais de atributo do MRO^{*} porém nesta figura não estão presentes todas as características de peças do MRO^{*} tal como na figura 4.3 do meta-esquema do MRO^{*}.

Definiu-se um relacionamento denominado "afeta" (com oposto "afetado por") entre tempo e propriedade para representar o fato de que um atributo temporal temporiza aquela propriedade. Este relacionamento tem como atributos as propriedades semânticas granularidade, densidade, e se o dado temporal é do tipo intervalo ou evento.

Como abordado na seção 5.3.2, objetos e relacionamentos tornam-se passíveis de tratamento temporal, devido a opção de adotar a abordagem atributo temporizado para controlar tempo. O relacionamento "atributos são" entre atributo e objeto, e atributo e relacionamento (binário e triplo) representa o tratamento temporal a objetos e relacionamentos.

Figura 5.7 - Caminho que determina os valores de um atributo temporizado

A propriedade semântica tempo de vida é representada pelo relacionamento "tempo de vida do" entre tempo e objetos, e tempo e relacionamentos.

5.3.7 - Exemplo de Definição de Informações Temporais.

Um exemplo de definição de atributo temporizado é apresentado no Apêndice A. A representação gráfica no DRO e no DRI é apresentada nas figuras 5.2 e 5.3¹¹.

5.4 - Determinação de Valores de um Atributo Temporizado.

No MRO*, o tempo que afeta um atributo temporizado pode ser composto por mais do

¹¹Os atributos temporais tempo válido e tempo da transação são pré-definidos.

que um atributo temporal.

A figura 5.7 apresenta um caminho que determina os valores de um atributo temporizado, segundo dois atributos temporais. Os valores v_i do atributo são determinados pelos atributos temporais associados ao atributo, no caso t_1 e t_2 em instantes específicos para cada um dos atributos temporais. Por exemplo, o valor v_3 é determinado pelo atributo temporal t_1 no instante 3 (t_{1-3}), e o atributo temporal t_2 no instante 1 (t_{2-1}). Isto consiste numa generalização do conceito de sequências de tempo, pois tratam n ($n \geq 1$) atributos temporais, enquanto que as sequências de tempo propostas em [SEGE87] tratam apenas um atributo temporal.

Deste modo o valor v_m de um atributo temporizado que esteja associado a n ($n > 0$) atributos temporais, é determinado por $t_{1-i}, t_{2-j}, \dots, t_{n-h}$ (vide figura 5.8). Os instantes $i, j, \dots, h$ não são necessariamente iguais, devido ao fato de ser possível realizarem-se operações passadas e futuros (retroativas e proativas) na BD e não apenas as operações referentes ao presente. Ou seja, na BD o tempo não flui continuamente.

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	.	.	v_m
t_{1-1}	t_{1-2}	t_{1-2}	t_{1-2}	t_{1-3}	t_{1-4}	t_{1-5}			.	.	t_{1-i}
t_{2-1}	t_{2-1}	t_{2-2}	t_{2-3}	t_{2-4}	t_{2-5}	t_{2-5}			.	.	t_{2-j}
t_{3-1}	t_{3-1}	t_{3-1}	t_{3-1}	t_{3-2}	t_{3-3}	t_{3-4}			.	.	t_{3-k}
t_{4-1}	..	..									.
t_{5-1}	.	.									.
.	.	.									.
.											.
t_{y-1}	t_{y-2}	t_{y-3}	t_{y-4}	t_{y-5}	t_{y-6}	t_{y-7}	t_{y-8}	t_{y-8}			.
t_{n-1}	t_{n-2}	t_{n-3}	t_{n-4}	t_{n-2}	t_{n-3}	t_{n-5}	t_{n-6}	t_{n-5}			t_{n-h}

Figura 5.8 - Valor de um atributo temporizado por vários atributos temporais

Quando um valor é adicionado, em geral ao menos um índice de um atributo temporal cresce em uma unidade.

5.5 - Considerações Finais.

Neste trabalho incorporaram-se recursos ao MRO* para a modelagem de informações temporais. Isso foi realizado incorporando-se ao MRO* os conceitos de tempo identificados como fundamentais para a classificação adotada apresentada no capítulo dois.

No próximo capítulo realizam-se considerações sobre a LAMRO - Linguagem de Acesso ao MRO*.

Capítulo 6

LAMRO* (Extensão)

6.1 - Considerações Iniciais.

Neste capítulo apresenta-se a linguagem de acesso ao MRO* - LAMRO*, definida anteriormente em [PIZZ92]. Para adicionar os conceitos temporais incorporados ao MRO* apresentados neste trabalho foram realizadas alterações em alguns comandos da LAMRO* e incluídos outros. Neste trabalho não são descritos todos os comandos da linguagem LAMRO*, mas somente os novos comandos e os que sofreram alguma alteração em sua concepção original.

Essa linguagem possui comandos para a manipulação tanto da BD propriamente dita quanto do esquema de dados. No MRO* considera-se que o esquema da aplicação evolui através de sucessivas versões (atemporais) que podem ser reativadas a critério do usuário. Como um esquema é tratado como uma colônia da base - composta de objetos, atributos, etc - tal como todo o restante da base (meta-sistema), esses elementos do esquema passam também a poder ser temporizáveis. Este trabalho, e em particular a implementação descrita no capítulo seguinte, permitem que se temporize os atributos dos objetos e relacionamentos de um esquema. No entanto o significado disso não foi analisado, embora reconheça-se que o impacto que isso traz sobre os conceitos de definição e evolução de esquemas precisam ser estudados em particular vinculado ao conceito de gerenciamento de versões de colônias em geral.

No MRO*/GEO a utilização de atributos temporais em cláusulas subsequentes é possível pois o modelo armazena TEMPO diretamente (as informações derivadas de operações realizadas com informações temporais armazenadas na base manterão a informação sobre tempo e podem ser utilizadas em operações subsequentes que manipulem tempo).

6.2 - Especificações.

Para possibilitar uma melhor definição e entendimento do subconjunto de comandos para a definição e manipulação de dados temporais algumas especificações foram desenvolvidas e são descritas a seguir.

6.2.1 - Classificação dos Comandos.

Os comandos da linguagem¹² atuam sobre os elementos do MRO* e suas instâncias tais como objeto, relacionamento e atributo. Baseados na função e finalidade a que se destinam os comandos foram classificados da seguinte forma:

- a) Inclusão - permite incluir a definição de um novo tipo ou a instância de um elemento do modelo;
- b) Eliminação - permite eliminar algum elemento anteriormente incluído;
- c) Substituição - permite a alteração de alguns dos elementos da base de dados;
- d) Estabelecimento de Ambiente - permite considerar colônias a fim de tornar objetos disponíveis e a manipulação de tempo associado aos componentes do MRO* passíveis de tratamento temporal.
- e) Seleção de Objetos - permite obter agrupamentos de objetos através de consultas a base de dados;
- f) Lista/máscara para Seleção - permite listar objetos pertencentes a uma seleção exibindo as informações através de um filtro - máscara;
- g) Mostra Elemento - permite exibir informações sobre os diferentes elementos pertencentes ao modelo.

Como ilustração da linguagem, alguns comandos são apresentados na seção 6.3 tal como descrito em [PIZZ92]. Na seção 6.4 apresenta-se as modificações que a linguagem¹³ sofreu para tratar atributos temporais.

6.2.2 - Notação Utilizada.

Na definição sintática dos comandos foi utilizada a seguinte notação:

As **palavras reservadas** da linguagem aparecem em letras maiúsculas e todos os argumentos fornecidos pelo usuário em letras minúsculas;

¹²Os comandos da linguagem foram divididos em um grupo que manipula o esquema e outro que manipula a base. De maneira geral os comandos de manipulação do esquema não afetam o estado corrente da base.

¹³Os valores de tempo limite inferior e superior que aparecem na sintaxe dos comandos a seguir não se aplicam ao atributo temporal tempo da transação para operações de atualização da BD pois apenas o SGBD atualiza os valores do tempo da transação.

< > delimitam um símbolo fornecido pelo usuário;

[] delimitam um conjunto de símbolos opcionais;

()^{*} delimitam um conjunto de símbolos que podem aparecer zero ou mais vezes;

()⁺ delimitam um conjunto de símbolos que podem aparecer uma ou mais vezes;

{ } delimitam um conjunto de símbolos dos quais ao menos um componente deve aparecer;

; finaliza um comando;

/* */ delimita um comentário.

Note que a linguagem é sensível ao caixa das letras para os símbolos definidos pelo usuário porém não às palavras reservadas.

6.3 - Exemplos da Linguagem.

Os comandos de inclusão permitem que elementos do MRO^{*}, possam ser introduzidos tanto na base como no esquema (tais como objetos, relacionamentos e atributos). Para que um relacionamento ou um atributo seja incluído na base deve haver um objeto corrente. Quando um objeto ou um relacionamento é criado na base torna-se corrente. De um modo geral, o comando "LOCALIZA" é utilizado para tornar um objeto, relacionamento ou colônia corrente na base.

Através dos comandos de eliminação os elementos do MRO^{*} incluídos tanto no esquema como na BD podem ser removidos. Sua sintaxe é semelhante aos comandos de inclusão para seus respectivos elementos utilizando o símbolo "ELIMINA".

6.3.1 - Operações de Elementos do MRO^{*} no Esquema.

O comando "DEFINE" seguido do nome do elemento é utilizado para incluir um elemento do modelo no esquema corrente. Este comando não afeta o estado corrente. A sintaxe deste comando é:

```
DEFINE TIPO <elemento> ...[:]
```

Qualquer elemento já definido no esquema pode ser referenciado em um comando de eliminação acrescentando em seguida ao comando, o símbolo "TIPO".

Um comando de eliminação de elementos do esquema só terá sucesso se não existirem ocorrências do elemento na BD. A sintaxe deste comando é:

ELIMINA TIPO <elemento> ...[:]

6.3.1.1 - Inclusão de Objeto.

Através do comando descrito a seguir é possível criar Tipos de Objetos em Esquemas de dados.

```
DEFINE OBJETO <tipo-objeto>
  HABITA (<tipo_colônia>)+
  [ IDENTIFICADO POR      <modalidade_grau1>
                        <tipo_obj_origem>
                        <tipo_obj_destino> ] [:]
```

A cláusula "IDENTIFICADO POR" é utilizada para definir que o tipo de objeto é identificado por um tipo de relacionamento.

6.3.1.2 - Inclusão de Atributo.

Através do comando descrito a seguir é possível criar Tipos de Atributos.

```
DEFINE <característica> <tipo_atributo>
  [TIPO DE DADO <tipo_dado> ]
  ASSOCIADO [A]
  ({ [RELACIONAMENTO <modalidade_grau1>
                        <tipo_objeto_origem>
                        <tipo_objeto_destino> ]
    [OBJETO      <tipo_objeto> ]
    [TRIPLO      <modalidade_grau2>
                  <tipo_objeto_d1>
                  <tipo_objeto_d2>
                  <tipo_objeto_d3> ]}]
  [ MÍNIMO <número> ]
  [ MÁXIMO <número> ])+
  [ INICIALIZAÇÃO ];
```

Um mesmo tipo de atributo pode estar associado a diferentes objetos e/ou relacionamentos.

A característica e o tipo de dado devem ser explicitados sempre que um tipo de atributo for definido pela primeira vez. O tipo de dado somente é especificado se a característica for propriedade.

6.3.1.3 - Eliminação de Atributo.

Através do comando descrito a seguir, é possível eliminar Tipos de Atributos.

```
ELIMINA TIPO ATRIBUTO <tipo_atributo>
[ASSOCIADO [A]
( [RELACIONAMENTO <modalidade_grau1>
    <tipo_objeto_origem>
    <tipo_objeto_destino> ) ]
( [OBJETO <tipo_objeto> ]* ]
( [TRIPLO <modalidade_grau2>
    <tipo_objeto_d1>
    <tipo_objeto_d2>
    <tipo_objeto_d3> ) ]* ]
];
```

Um tipo de atributo pode ser eliminado de cada tipo de elemento individualmente através da cláusula "ASSOCIADO A" ou pode-se eliminar sua definição desde que não esteja associado a nenhum tipo de elemento, apenas especificando o tipo de atributo.

6.3.2 - Operações de Elementos do MRO* na Base de Dados.

Para que um elemento possa ser incluído na BD o seu tipo deve estar pré-definido no esquema. De forma geral, o comando utilizado para incluir um elemento (atributo, objeto, etc) na BD é composto pelo nome do elemento que será incluído (por exemplo, atributo), seguido pelo seu tipo (por exemplo, tipo do atributo) e pelo próprio elemento. A sintaxe deste comando é apresentada nas próximas seções de acordo com as variações necessárias.

O comando "ELIMINA" permite a remoção de qualquer informação existente na BD. O comando segue o mesmo padrão anterior apresentado na seção 6.3.1 a menos da ausência do símbolo "TIPO".

6.3.2.1 - Inclusão de Objetos.

A inclusão de objetos na BD é feita na colônia corrente de tipo habitado pelo tipo de objeto. Após este comando o objeto será o corrente. A sintaxe é a seguinte:

```
OBJETO <t_objeto> (<nome_objeto>)* [;]
OBJETO <t_objeto> IDENTIFICADO POR RELACIONAMENTO [;]
```

Um objeto já incluído na BD pode ser localizado através do comando "LOCALIZA":

```
LOCALIZA OBJETO <t_objeto> <nome_objeto> [;]
```

LOCALIZA OBJETO <t_objeto> IDENTIFICADO POR RELACIONAMENTO [;]

6.3.2.2 - Inclusão de Valor de Atributo.

O valor do atributo a ser incluído refere-se ao objeto ou relacionamento corrente. Desta forma pode-se indicar a qual elemento do modelo a inclusão de atributo está associada e caso o elemento não esteja explicitado será tomado o último elemento localizado ou incluído na base. A sintaxe é a seguinte:

ATRIBUTO [<elemento>] <t_atributo> <valor> (<valor>)* [;]

6.3.2.3 - Eliminação de Valor de Atributo.

Valores de atributo associados a um elemento (objeto/relacionamento) que deve estar corrente podem ser eliminados individualmente (especificando os valores) ou todas as ocorrências do atributo (especificando somente seu tipo). A sintaxe é a seguinte:

ELIMINA ATRIBUTO [<elemento>] <t_atributo> [<valor> (<valor>)*] [;]

6.4 - Extensão da Linguagem Temporal.

Nesta seção apresenta-se a linguagem de acesso ao MRO* - LAMRO* com as alterações e extensões realizadas, que tornaram-na uma linguagem temporal.

Em decorrência da política de não-eliminação em BDTs, dados temporizados não poderão ter as suas definições eliminadas do esquema, desde que ao menos uma ocorrência tenha sido armazenada na BD. Em caso de eliminação de dados temporizados na BD estes serão eliminados apenas logicamente.

6.4.1 - Operações de Elementos do MRO* no Esquema.

6.4.1.1 - Inclusão de Atributo Temporal.

Quando um atributo for temporizado, primeiro deve-se definir a característica TEMPO seguida pelo atributo temporal que se deseja definir (caso seja um atributo temporal definido pelo usuário, pois, tempo válido e tempo da transação são pré-definidos e reconhecidos pelo SGBD). A sintaxe deste comando é a seguinte:

DEFINE TEMPO <atributo_temporal> [;]

6.4.1.2 - Inclusão de Atributo Temporizado.

Após a definição do atributo temporal, define-se o atributo temporizado, o tipo de dado seguido da cláusula AFETADO POR seguida de um ou mais atributos temporais. Se o atributo temporal referenciado não existir é emitida uma mensagem de erro. Após o atributo temporal, especificam-se as propriedades semânticas granularidade, densidade e se o atributo temporizado é do tipo intervalo ou evento. Se a granularidade e/ou densidade definidas não forem válidas é emitida uma mensagem de erro. Caso a granularidade e a densidade definidas sejam incompatíveis com o que foi definido anteriormente para outros atributos temporizados associados ao objeto ou relacionamento, emite-se uma mensagem de erro. A sintaxe deste comando é a seguinte:

```

DEFINE PROPRIEDADE14 <atributo temporizado>
[TIPO DE DADO <tipo_dado> ]
[AFETADO POR (<atributo_temporal> (<atributo_temporal>)*)+] [:]
(
  [GRANULARIDADE <unidade>]           /* dia */
  [DENSIDADE <densidade>]             /* passo constante */
  [INTERVALO|EVENTO]                  /* intervalo */
)
ASSOCIADO [A]
({
  [RELACIONAMENTO <modalidade_grau1>
    <tipo_objeto_origem>
    <tipo_objeto_destino>]
  [OBJETO <tipo_objeto>]
  [TRIPLO <modalidade_grau2>
    <tipo_objeto_d1>
    <tipo_objeto_d2>
    <tipo_objeto_d3>]]
[ MÍNIMO <número> ]
[ MÁXIMO <número> ])+
[ INICIALIZAÇÃO ];
  
```

A ação semântica correspondente é:

- 1- Criar o Objeto <atributo temporizado> e seu atributo <característica>;
- 2- Enquanto houver <atributo temporal> no comando faça (3 até 6);
- 3- Procurar o <atributo temporal>;
- 4- Se não encontrou então operação inválida;
- 5- Criar o Relacionamento AFETA entre os Objetos:
<atributo temporal> e <atributo temporizado>;

¹⁴Neste trabalho realizou-se somente a temporização da característica Propriedade.

- 6- Criar os atributos *Granularidade*, *Intervalo/Evento*, *Densidade* no Relacionamento *AFETA*;
- 7- Se <característica> é Sinônimo então faça (8 até 11);
- 8- Enquanto houver <tipo_objeto> no comando faça (9 até 11);
- 9- Procurar o <tipo_objeto>;
- 10- Se não encontrou então operação inválida;
- 11- Criar o relacionamento *IDENTIFICA* entre os Objetos:
<atributo temporizado> e <tipo_objeto>;
- 12- Se <característica> não é Sinônimo então faça (13 até 29);
- 13- Enquanto houver <tipo_objeto> no comando faça (14 até 19);
- 14- Procurar o <tipo_objeto>;
- 15- Se não encontrou então operação inválida;
- 16- Criar o relacionamento *ATRIBUTO DE* entre os Objetos:
<atributo temporizado> e <tipo_objeto>;
- 17- Criar os atributos *Multiplicidade Mínima* e *Máxima* no Relacionamento *ATRIBUTO DE*;
- 18- Se a <característica> é Propriedade então faça 19;
- 19- Criar o atributo <tipo_dado> no Objeto <atributo temporizado>;
- 20- Enquanto existirem Tipos de Relac. Binários¹⁵ no comando faça (21 até 24);
- 21- Procurar o Tipo de Relac. Binário;
- 22- Se não encontrou então operação inválida;
- 23- Criar o relac. *ATRIBUTO DE* entre os Objetos:
<atributo temporizado> e Tipo de Relac. Binário;
- 24- Criar os atributos *Multiplicidade Mínima* e *Máxima* no relacionamento *ATRIBUTO DE*;
- 25- Enquanto existirem Tipos de Relac. Triplos¹⁶ no comando faça (26 até 29);
- 26- Procurar o Tipo de Relac. Triplo;
- 27- Se não encontrou então operação inválida;
- 28- Criar o relac. *ATRIBUTO DE* entre os Objetos:
<atributo temporizado> e Tipo de Relac. Triplo;
- 29- Criar os atributos *Multiplicidade Mínima* e *Máxima* no relacionamento *ATRIBUTO DE*;
- 30- Fim;

Exemplo: DEFINE PROPRIEDADE Total

¹⁵Um Tipo de Relacionamento Binário é especificado pelos Tipos de Objeto Origem e Destino que o utilizam e pela Modalidade Grau 1 que o compõe.

¹⁶Um Tipo de Relacionamento Triplo é especificado pelos Tipos de Objeto da Divisão 1, Divisão 2 e Divisão 3 que o utilizam e pela Modalidade Grau 2 que o compõe.

TIPO DE DADO Real
AFETADO POR Tempo_Válido, Tempo_Transação
ASSOCIADO OBJETO Informação_Financeira;

No apêndice A apresenta-se a definição de atributos temporizados para o exemplo da aplicação de informações financeiras. A figura 5.2 apresenta o DRO correspondente ao exemplo.

6.4.1.3 - Eliminação de Atributo Temporal.

Este comando elimina um atributo temporal associado a um esquema.

Sintaxe:

ELIMINA TEMPO <atributo_temporal> [;]

6.4.1.4 - Eliminação de Atributo Temporizado.

Este comando possui algumas variações que são apresentadas a seguir. Quando especifica-se **ELIMINA PROPRIEDADE** <atributo_temporizado> o atributo é eliminado do esquema. Caso o atributo não esteja definido no esquema uma mensagem de erro é emitida. Quando especifica-se **AFETADO POR** somente a associação entre o atributo temporal e o atributo temporizado é desfeita. A associação entre um atributo temporizado e objetos e/ou relacionamentos pode ser desfeita especificando-se a cláusula **ASSOCIADO** A seguida do objeto ou relacionamento. Quando emite-se um comando de eliminação de um atributo temporal ou temporizado que não esteja definido uma mensagem é enviada informando esta situação. A sintaxe deste comando é a seguinte:

```
ELIMINA PROPRIEDADE <atributo_temporizado>
  (AFETADO POR (<atributo_temporal>)*)* [;]
ASSOCIADO [A]
  ({    [RELACIONAMENTO <modalidade_grau1>
        <tipo_objeto_origem>
        <tipo_objeto_destino>]
        [OBJETO      <tipo_objeto>]
        [TRIPLO      <modalidade_grau2>
        <tipo_objeto_d1>
        <tipo_objeto_d2>
        <tipo_objeto_d3> ]}]
  [ MÍNIMO <número> ]
  [ MÁXIMO <número> ])+
  [ INICIALIZAÇÃO ];
```

A ação semântica correspondente é:

- 1- Enquanto houver <atributo_temporal> no comando faça (2);
- 2- Remover os Relacionamentos envolvendo o <atributo_temporal> com o <atributo_temporizado>;
- 3- Enquanto houver <tipo_objeto> no comando faça (4);
- 4- Remover os relacionamentos envolvendo o <tipo_objeto> com o <atributo_temporizado>;
- 5- Enquanto existirem Tipos de Relac. Binários no comando faça (6);
- 6- Remover os relacionamentos envolvendo os Tipos de Relac. Binário indicados com o <atributo_temporizado>;
- 7- Enquanto existirem Tipos de Relac. Triplos no comando faça (8);
- 8- Remover os relacionamentos envolvendo os Tipos de Relac. Triplo indicados com o <atributo_temporizado>;
- 9- Se não existirem relac. envolvendo o <atributo_temporizado> então faça (10);
- 10- Remover o Objeto <atributo_temporizado>;
- 11- Fim;

Exemplo: ELIMINA PROPRIEDADE Total
AFETADO POR Tempo_Válido, Tempo_Transação
ASSOCIADO OBJETO Informação_Financeira;

6.4.2 - Operações de Elementos do MRO* na Base de Dados.

Em caso de inclusão (eliminação) dos componentes do MRO* passíveis de tratamento temporal (objetos e relacionamentos) o tempo corrente é atribuído ao limite inferior (superior) do tempo de vida do atributo temporal tempo da transação, no caso do tempo da transação estar associado ao componente.

A eliminação de um objeto ou relacionamento temporizado implica na atribuição de valores de tempo ao limite superior dos atributos temporais e do tempo de vida associados aos atributos temporizados do objeto ou relacionamento.

Para cada atributo temporal a operação de inclusão/exclusão adiciona/encerra intervalos ou pontos de tempo a uma reta que representa o atributo temporal.

Os atributos temporais (inclusive a propriedade tempo de vida) podem ter os seus valores atualizados pelo usuário. A única exceção é o tempo da transação cujos valores são atualizados apenas pelo SGBD sendo que os usuários podem somente consultá-los.

6.4.2.1 - Inclusão de Valor de Atributo Temporizado.

O valor do atributo temporizado é incluído associado aos valores de tempo (limite inferior e superior) dos atributos temporais. Quando o atributo temporizado for do tipo

intervalo especifica-se os valores de tempo utilizando-se as "cláusulas" **DE limite_inferior** **A limite_superior**. Quando o atributo temporizado for do tipo evento, especifica-se o valor de tempo utilizando-se a "cláusula" **EM limite_inferior** ressaltando-se que neste caso o limite_inferior indica quando ocorreu o evento.

A inclusão de um atributo temporizado implica na atribuição de valores de tempo ao limite inferior e opcionalmente ao limite superior dos atributos temporais associados ao atributo. A sintaxe deste comando é a seguinte:

```
ATRIBUTO[<elemento>]<atributo_temporizado>(<valor>(<atributo_temporal>
    DE limite_inferior [A limite_superior])))+
ou
ATRIBUTO[<elemento>]<atributo_temporizado>(<valor>(<atributo_temporal>
    EM limite_inferior))*)+
```

Exemplo: ATRIBUTO Total 5010100,92
EM '19.11.1992'

6.4.2.2 - Eliminação de Valor de Atributo Temporizado.

O valor do atributo temporizado é eliminado associado aos valores de tempo (limite inferior e superior) dos atributos temporais. Quando o atributo temporizado for do tipo intervalo, especificam-se os valores de tempo utilizando-se as "cláusulas" **DE limite_inferior** **A limite_superior**. Quando o atributo temporizado for do tipo evento, especifica-se o valor de tempo utilizando-se a "cláusula" **EM limite_inferior**, ressaltando-se que neste caso o limite_inferior indica quando ocorreu o evento. Quando emite-se um comando de eliminação de um atributo temporizado cujo valor e os limites inferior e superior não correspondem a alguma ocorrência na BD emite-se um mensagem informando esta situação.

A eliminação de um atributo temporizado implica na atribuição de valores de tempo ao limite superior dos atributos temporais associados ao atributo. A sintaxe deste comando é a seguinte:

```
ELIMINA ATRIBUTO [<elemento>] <atributo_temporizado>
    (<valor> (<atributo_temporal>
    [DE limite_inferior A limite_superior])))+
ou
ELIMINA ATRIBUTO [<elemento>] <atributo_temporizado>
    (<valor> (<atributo_temporal>
    [EM limite_inferior])))+
```

Exemplo: ELIMINA ATRIBUTO Total 5010100,92
EM '19.11.1992'

6.4.3 - Operações do Tempo de Vida.

Quando algum objeto ou relacionamento é temporizado o tempo de vida é automaticamente associado a ele.

Devido a opção adotada neste trabalho (vide seção 5.3.4.2) necessita-se definir as operações de inclusão e substituição para que seja possível manipular os valores do tempo de vida. A operação de eliminação não se aplica devido a solução adotada neste trabalho. A solução adotada foi a representação do tempo de vida através de um registro e não de uma lista.

As operações de manipulação do tempo de vida são apresentadas a seguir.

6.4.3.1 - Inclusão de Valores do Tempo de Vida.

Os valores do tempo de vida são inclusos para algum elemento (objeto ou relacionamento), e referem-se a um ou mais atributos temporais. A sintaxe deste comando é a seguinte:

```
TEMPO DE VIDA <elemento> (<atributo_temporal>  
DE limite_inferior A limite_superior)+
```

6.4.3.2 - Substituição de Valores do Tempo de Vida.

Os valores do tempo de vida são substituídos em algum elemento (objeto ou relacionamento), e referem-se a um ou mais atributos temporais. A sintaxe deste comando é a seguinte:

```
SUBSTITUIÇÃO TEMPO DE VIDA <elemento> (<atributo_temporal>  
DE limite_inferior A limite_superior)+
```

6.5 - Substituição.

Normalmente uma substituição pode ser executada na BD do MRO* através de uma sequência de comandos classificados como de eliminação e inclusão. No entanto, a eliminação de algum elemento da BD com objetivo de uma substituição pode significar a perda não desejável de muitas outras informações por isso apenas alguns elementos do MRO* foram contemplados com esta classe de instrução. Em [PIZZ92] são apresentados os elementos contemplados com esta classe de instrução.

Contudo, esta classe de comandos não é afetada pelos aspectos temporais desenvolvidos neste trabalho. Assim, não é necessário o desenvolvimento desta classe de instrução para aspectos temporais.

6.6 - Recuperação.

Os comandos **MOSTRA** foram criados com o objetivo de exibir ao usuário o estado em que a base encontra-se num determinado momento.

Desta forma o comando deve ser versátil para possibilitar visualizar a base segundo seus diferentes elementos, tais como: colônia, objeto, relacionamento e atributo.

Os comandos terão sua funcionalidade estendida para tratar aspectos temporais, porém, neste trabalho apresenta-se somente os elementos do modelo que são afetados pelo tratamento de tempo.

De maneira geral, adotou-se as seguintes convenções para o comando mostra:

- Utilizou-se o "token" **HABITA** fazendo referência a objeto (ou tipo) pertencente a alguma colônia (ou tipo);
- O "token" **CONSTRITA** utilizou-se para referir-se a alguma colônia (ou tipo) constrita por alguma outra colônia (ou tipo);
- O token "**CORRENTE**", foi utilizado normalmente para os comandos referentes à Base.

6.6.1 - Objetos.

6.6.1.1 - Com relação ao Esquema.

Mostra os tipos de objetos¹⁷ existentes no esquema, o tipo de colônia a qual ele pertence, os tipos de relacionamentos para o qual é origem e os tipos de atributos associados - para atributos temporizados informa-se esta condição. A sintaxe é a seguinte:

MOSTRA TIPOS OBJETOS;

Para um tipo de objeto específico a sintaxe é:

MOSTRA TIPO OBJETO <tipo_objeto>;

6.6.1.2 - Com relação à Base.

Mostrar o nome e o tipo do objeto corrente sendo manipulado, o caminho de acesso para a colônia onde ele habita, a modalidade de relacionamento e objeto destino associado, os tipos e valores de atributos associados a ele. Para os atributos temporizados, serão mostrados os atributos temporais, seus valores de tempo e o tempo de vida. A sintaxe é a

¹⁷O comando **MOSTRA** manipula objetos da base definidos para as colônias e pode-se obter os valores de seus atributos através de variantes do comando definidas para atributos como será visto nas seções posteriores.

seguinte:

MOSTRA OBJETO CORRENTE;

6.6.2 - Relacionamentos.

6.6.2.1 - Com relação ao Esquema.

Este comando exibe os tipos de relacionamentos binários existentes para cada modalidade de relacionamento e para cada tipo mostra o nome da modalidade de relacionamento, o nome da modalidade de relacionamento oposto, o tipo de objeto origem e o tipo de objeto destino. A sintaxe é a seguinte:

MOSTRA TIPOS RELACIONAMENTOS;

Para uma modalidade de relacionamento este comando mostra o nome do tipo de relacionamento, o nome do tipo de relacionamento oposto, o tipo de objeto origem o tipo de objeto destino, a multiplicidade mínima, a multiplicidade máxima, a multiplicidade mínima oposta, a multiplicidade máxima oposta, opção por segurança ("SIM/NÃO") e os tipos de atributos associados - para atributos temporizados informa-se esta condição. A sintaxe é a seguinte:

MOSTRA TIPO RELACIONAMENTO <modalidade>;

Uma outra variante do comando exibe os tipos de relacionamentos triplos do esquema com as seguintes informações: o nome do tipo de relacionamento triplo, o nome da modalidade da divisão dois, o nome da modalidade da divisão três, o nome do tipo de objeto pertencente a divisão um, o nome dos tipos de objetos pertencentes a divisão dois e o nome do tipo de objeto pertencente a divisão três. A sintaxe é a seguinte:

MOSTRA TIPOS RELACIONAMENTOS TRIPLOS;

Outra variante do comando permite consultar todos os tipos de relacionamentos triplos existentes no esquema sendo manipulado, e para cada um, mostra as seguintes informações: o nome do tipo de relacionamento triplo, a multiplicidade máxima do relacionamento triplo, a multiplicidade mínima do relacionamento triplo; o nome da modalidade que compõe a divisão dois, seu nome correspondente atribuído na definição do triplo, a multiplicidade máxima do relacionamento da divisão, a multiplicidade mínima do relacionamento da divisão; o nome da modalidade que compõe a divisão três, seu nome correspondente atribuído na definição do triplo; nome do tipo de objeto pertencente a divisão um; nome do tipo de objeto pertencente a divisão dois; opção por segurança ("SIM/NÃO"). A sintaxe é a seguinte:

MOSTRA RELACIONAMENTOS TRIPLOS <nome_relacionamento>;

6.6.2.2 - Com relação à Base.

Este comando exibe as seguintes informações do relacionamento binário corrente: tipo e nome dos objetos origem e destino, e a modalidade do relacionamento; valor e tipo dos atributos associados ao relacionamento; para os atributos temporizados são mostrados o atributo temporal e seus valores de tempo. A sintaxe é a seguinte:

MOSTRA RELACIONAMENTO CORRENTE;

Outra variante do comando exibe as informações do relacionamento triplo corrente: nome e tipo do objeto da divisão 1, modalidade do relacionamento da divisão 2, modalidade do relacionamento da divisão 3, nome e tipo do objeto da divisão 3, tipo e valor dos atributos do relacionamento triplo; para os atributos temporizados são mostrados os atributos temporais e seus valores de tempo. A sintaxe é a seguinte:

MOSTRA TRIPLO CORRENTE;

6.6.3 - Atributos.

6.6.3.1 - Com relação ao Esquema.

Este comando exibe as seguintes informações dos tipos de atributos definidos no esquema corrente: nome do atributo, característica, tipo de dado e se é temporizado. A sintaxe é a seguinte:

MOSTRA TIPOS ATRIBUTOS;

Outra variante do comando permite exibir as seguintes informações sobre um tipo de atributo: nome do atributo, característica, tipo de dado, multiplicidade mínima, multiplicidade máxima, nome de tipos de objetos ou relacionamentos associados; se temporizado, mostra os atributos temporais aos quais está associado. A sintaxe é a seguinte:

MOSTRA TIPO ATRIBUTO <tipo_atributo>;

6.6.3.2 - Com relação à Base.

Para um objeto ou relacionamento corrente, este comando exibe as seguintes informações: valores associados ao tipo de atributo para o elemento corrente,

se temporizado, mostra os atributos temporais e seus respectivos valores. A sintaxe é a seguinte:

```
MOSTRA ATRIBUTO [<elemento>] <tipo_atributo>;
```

6.7 - Comando LISTA.

O comando "LISTA" permite listar o nome dos objetos que participam de uma seleção. Seu formato geral é:

```
LISTA <seleção>;
```

Máscara pode ser entendida como um filtro aplicado na seleção. Máscara define um conjunto de tipos de atributos e modalidades de relacionamentos, que já foram definidos previamente no esquema. Pode-se criar uma máscara, onde o conjunto de tipos de atributos e relacionamentos, bem como sua ordem, são indicados. A sintaxe do comando é a seguinte:

```
DEFINE MASCARA <nome> '{'
    (<tipo_atributo> | <modalidade_relacionamento>)* '}'
    [ORDEM CRESCENTE|DECRESCENTE
        (<tipo_atributo> | <modalidade_relacionamento>)]
    [REPETIÇÃO SIM|NÃO]; /* Assume SIM*/
```

Uma máscara permite que se manipule as informações relativas a objetos, ou relacionamentos, cujos atributos deseja-se indicar. Na definição de uma máscara é admitida a especificação de ordem e/ou repetição com que os dados mascarados possam ser utilizados. Máscaras são utilizadas juntamente com seleções, criando-se tabelas de objetos, onde os atributos da máscara constituem os atributos e os relacionamentos, aos quais se reporta a operação em andamento. Para a utilização das máscaras, já definidas ou não, deve-se acrescentar a cláusula "MASCARA", do comando "LISTA", no seguinte formato:

```
LISTA <seleção> MASCARA <nome>
QUANDO (<atributo_temporal> [DE limite_inferior A limite_superior])+
    (<atributo_temporal> [EM limite_inferior])+ ;
ou
LISTA <seleção> MASCARA '{'
    (<atributo> | <relacionamento>)* '}'
    [ORDEM CRESCENTE|DECRESCENTE
        (<tipo_atributo> | <modalidade_relacionamento>)]
    [REPETICAO SIM|NAO] /*assume SIM*/
QUANDO (<atributo_temporal> [DE limite_inferior A limite_superior])+
    (<atributo_temporal> [EM limite_inferior])+ ;
```


Na seção 6.7.1 aborda-se a cláusula "QUANDO" que foi adicionada ao comando para tratar aspectos temporais do MRO*.

6.7.1 - Cláusula QUANDO.

Esta cláusula consiste da palavra chave "QUANDO" seguida por um ou mais atributos temporais e as restrições aos valores de tempo de interesse para cada um dos atributos temporais especificados. Quando o valor de tempo a ser informado for intervalo, este é expresso pelas variantes **DE limite_inferior A limite_superior**. Quando tratar-se de um evento é expresso pela variante **EM limite_inferior**. Quando os valores de tempo não forem informados, o resultado será um subconjunto com 0, 1 ou mais valores e seus respectivos valores de tempo associados.

Considerou-se que recursos do tipo "begin of", "end of", "overlap", "extent" e outros poderiam resultar em maiores facilidades para o usuário, porém, aumentaria o número de comandos da linguagem o que não é desejável. As operações realizadas pelo usuário com comandos, tais como, "begin of", "end of", podem ser realizadas sem eles ao custo de um número maior de comandos para expressar uma consulta, sendo esta a opção adotada.

Exemplo : Suponha que deseja-se saber qual é o total de depósitos a vista na unidade organizacional de São Carlos em 19.11.1992.

```
LOCALIZA uniorg São_Carlos  
LISTA (Possui Informação_financeira = depósito_vista)  
MASCARA (Total, Informação_financeira, tempo_válido)  
QUANDO tempo_transação EM '19.11.1992'
```

O resultado desta consulta é:

```
Totalização da Informação Financeira : Cr$5.010.100,92  
Informação Financeira : Depósito a Vista  
Tempo Válido : 19.11.92
```

6.8 - Estabelecimento de Ambiente.

6.8.1 - Estabelecimento de Sessão.

Para que o usuário inicie uma sessão em uma BD MRO*, este deve ser reconhecido pelo sistema, bem como tornar seu esquema e sua base de trabalho ativos. Isto é feito através do seguinte comando.

Sintaxe:

```
ESQUEMA <nome_do_esquema> BASE <nome_da_base_de_dados>
```

USUÁRIO <nome_do_usuario> [;]

O usuário finaliza uma sessão na BD MRO* através do comando:

Sintaxe:

FIM [;]

6.8.2 - Valores do Ambiente.

Para cada atributo temporal definido na BD devem existir dois valores de tempo definidos : **Real** e **Assumido**. O valor **Real** é aquele que considera o tempo no mundo real. O valor **Assumido** é o que é usado para a manipulação da BD.

Antes de realizar alguma operação temporal, o usuário deve atribuir valores de tempo (limite inferior e superior) aos atributos temporais do tempo Real e do Assumido, caso seja de seu interesse. A sintaxe do comando é a seguinte:

TEMPO <atributo_temporal> [REAL|ASSUMIDO]
DE limite_inferior A limite_superior [;]
EM limite_inferior [;]

Para cada atributo temporal definido na BD existirão as variáveis correspondentes (limites inferior e superior) para os valores **REAL** e **ASSUMIDO**.

O tempo Real pode ser redefinido pelo usuário, e é atualizado pelo relógio da máquina, sendo mantido entre sessões diferentes do gerenciador, exceto para o atributo temporal tempo da transação.

O tempo Assumido pode ser redefinido pelo usuário, e não é atualizado pelo relógio da máquina. Não é mantido entre sessões diferentes do gerenciador, sendo igualado ao tempo Real no início de cada sessão. Pode-se igualar o tempo Assumido ao Real através do comando:

TEMPO <atributo_temporal> ASSUMIDO = REAL [;]

Neste caso o tempo Assumido é igualado ao tempo Real e passa a acompanhar o relógio da máquina até que seja redefinido pelo usuário, somente para o atributo temporal referenciado.

O atributo temporal tempo da transação utiliza o tempo Real para atualizações na BD e o tempo Assumido para consultas. Os demais atributos temporais utilizam o tempo Assumido para atualizações e consultas caso o tempo Assumido esteja definido. Senão utilizam o tempo Real o qual está sempre definido.

Pode-se obter os tempos Real e Assumido pelo gerenciador em um determinado momento através do comando:

MOSTRA TEMPO [REAL|ASSUMIDO] [;]

Atributo Temporal	REAL		ASSUMIDO	
t. válido	limite inferior	limite superior	limite inferior	limite superior
t.transação	"	"	"	"
t.1 def. p/o usuário	"	"	"	"
.	"	"	"	"
t.n def. p/o usuário	"	"	"	"

Figura 6.1 - Valores temporais do ambiente.

A figura 6.1 ilustra os conceitos descritos nesta seção.

6.8.3 - Ambiente Não Estabelecido.

Descreve-se a seguir o comportamento das operações de manipulação da BD quando o ambiente não estiver estabelecido. Essas considerações aplicam-se a todos os atributos temporais definidos na BD, exceto em alguns casos especiais referentes ao tempo da transação, mencionados na seção 6.7.2.

6.8.3.1 - Inclusão.

Assume-se que o limite inferior é o instante corrente ("now") e o limite superior é ∞ (infinito) para todos os atributos temporais definidos na BD.

6.8.3.2 - Consulta.

Análogo a inclusão.

6.8.3.3 - Exclusão.

Análogo a inclusão. Porém, os valores assumidos implicam que a operação de exclusão aplica-se somente a informações válidas no instante da execução ou no futuro.

6.9 - Considerações Finais.

Neste capítulo apresentou-se a Linguagem de Acesso ao MRO* - LAMRO* modificada para tratar os aspectos temporais incorporados ao MRO*.

O padrão definido para a LAMRO foi seguido, estendendo-se os comandos já existentes para tratar tempo; os novos comandos resumem-se ao mínimo.

O objetivo da linguagem, que é verificar a aplicabilidade dos conceitos numa linguagem completa foram atingidos, os conceitos foram exemplificados pela linguagem e a linguagem demonstra as operações que o SGBD deve satisfazer.

No próximo capítulo apresenta-se a descrição da implementação realizada.

Capítulo 7

Descrição da Implementação

7.1 - Considerações Iniciais.

Neste capítulo são abordados os aspectos da implementação realizada para a validação dos conceitos apresentados neste trabalho.

A implementação utiliza primitivas do Gerenciador de Objetos (GEO) desenvolvido no Instituto de Ciências Matemáticas de São Carlos - USP, o qual suporta parte dos conceitos do MRO*.

Foi utilizado um editor experimental denominado GEO INC, no qual todos os trabalhos envolvendo BD do GEO são implantadas e testadas antes de realmente comporem o núcleo do gerenciador e se tornarem disponíveis para utilização.

```
struct MROObjeto {
    codiide    nome;
    coditipo   tipo;
    coditipo   subtipo;
    codirel    relac[4];
    codirel    rel_identificador;
    codiatr    atrib;
    codicol    constrital;
    codiobjeto mestre;
    coditmp    tempovida;
    codiobjeto quem_criou,
               quem_mudou;
    pal2       estado;
    long        marca[MAX_MARCAS];
} MRO_objeto;
```

Figura 7.1 - Arquivo de Objetos

7.2 - Descrição dos Arquivos Lógicos.

Nesta seção apresenta-se a estrutura dos arquivos lógicos utilizados na implementação do GEO para representar os conceitos do MRO* Temporal.

Os elementos de uma BD GEO são armazenados fisicamente em um único arquivo de dados. Este arquivo físico contém as informações de todas as instâncias da BD. A grosso modo, pode-se dizer que a arquitetura do GEO é dividida em 2 camadas (vide figura 7.2). A primeira camada, denominada Núcleo gerencia o acesso ao arquivo físico da BD, atuando como um monitor para emular a existência de arquivos lógicos, cada um dispondo de métodos de acesso bem definidos, supridos pelo próprio núcleo. Essa camada é responsável também pelo gerenciamento de um cache em memória, controle de acesso multi-usuário, controle de log, etc.

A segunda camada, denominada Implementação do Modelo, provê as primitivas da Interface de Programação de Aplicativos, implementando os conceitos do MRO* utilizando os arquivos lógicos suportados pelo núcleo.

Assim, um arquivo físico é composto por vários arquivos lógicos. A BD MRO* encontra-

Figura 7.2 - Arquitetura do GEO

se armazenada nesses diversos tipos de arquivos lógicos. O Arquivo Lógico de Objetos é responsável por armazenar as informações sobre as instâncias dos objetos na BD como o seu tipo, o conjunto de identificadores, indicação dos relacionamentos que faz parte, tipos de atributos associados, colônia a que pertence, tempo de vida que restringe os valores de tempo aplicáveis ao objeto entre outros dados. Informações sobre os demais elementos da BD, como Relacionamentos, Colônias, Atributos, são armazenados nos respectivos arquivos lógicos, como por exemplo, o Arquivo de Relacionamentos, Arquivo de Atributos, etc.

É importante observar que como temos um Arquivo Lógico para cada conceito, deve-se tomar cuidado com a quantidade e o tipo da informação que irá definir o conteúdo de cada um. Para evitar a redundância dos dados da BD cada um de seus arquivos apresentam apenas as informações que sejam estritamente necessárias para caracterizar aquela peça dentre as demais. Como exemplo, considere o Arquivo de Objetos. Como foi observado anteriormente este armazena uma série de informações para que o objeto esteja completamente descrito. Mas os únicos dados que são efetivamente armazenados nesse arquivo é o Tipo do objeto. As demais informações como seu identificador principal, relacionamentos e atributos, encontram-se armazenadas em seus arquivos correspondentes. No arquivo de objetos há somente uma indicação para registros dos demais arquivos com a informação efetiva. Assim todos os arquivos lógicos contêm uma informação que seja

própria deste elemento, e um conjunto de indicadores para os demais arquivos com o restante das informações de descrição da peça armazenada.

Nas próximas seções descrevem-se os arquivos lógicos criados ou modificados, devido ao desenvolvimento deste trabalho.

7.2.1 - Estrutura de Dados Básica.

Na seção 5.3.3 foi definida a utilização de intervalo de tempo para a representação de tempo no MRO*. A estrutura de dados definida para representar o intervalo de tempo, composto por limite inferior e limite superior é v_tempo, apresentada a seguir.

struct vtempo { t_tempo inicio, fim; }
v_tempo;

A indicação *t_tempo* é o tipo de dado básico utilizado para representar qualquer valor de tempo na implementação realizada.

```
struct MRORelacionamento {
    coditipo    tipo;
    coditipo    subtipo;
    codirel     prox_rel,
                volta_rel,
                oposto,
                proxtripla;
    coditri     tripla;
    codiobjeto  destino,
                identifica;
    coditmp     tempovida;
    codiatr     atrib;
} MRO_relacionamento;
```

Figura 7.3 - Arquivo de Relacionamento

7.2.2 - Arquivo Lógico dos Componentes do MRO* Afetados Pelo Tempo.

Como abordado na seção 5.3.4.2 o tempo de vida está associado aos componentes do MRO* que são afetados pelo tempo. Os componentes são Objeto, Relacionamento Binário e Relacionamento Triplo. Nos arquivos de cada um desses componentes foram inseridos ponteiros (*tempovida*) que apontam para o registro que representa tempo de vida que é um registro com a mesma estrutura do registro de tempo, *MRO_tempo*. Esses ponteiros foram as únicas alterações realizadas nos arquivos dos componentes citados sendo que todas as outras informações que fazem parte desses componentes são motivadas por outros assuntos que estão fora do escopo deste trabalho. Deste modo, não são descritas as outras informações componentes dos arquivos de Objeto, Relacionamento Binário e Relacionamento Triplo. As figuras 7.1, 7.3, e 7.4 ilustram respectivamente a estrutura dos arquivos de Objeto, Relacionamento Binário, e Relacionamento Triplo.

7.2.3 - Arquivo Lógico de Propriedade.

O arquivo de propriedades foi alterado, incluindo-se nele um ponteiro (*reg_tempo*) que aponta para o registro de tempo. Isto se deve ao fato de neste trabalho ter-se optado por temporizar somente o atributo com característica de propriedade. Esse ponteiro foi a única

alteração realizada no arquivo de Propriedade, sendo que todas as outras informações deste arquivo são motivadas por outros assuntos que estão fora do escopo deste trabalho. Deste modo, não são descritas as outras informações componentes do arquivo de Propriedade. A figura 7.5 ilustra a estrutura do arquivo de Propriedade.

7.2.4 - Arquivo Lógico de Tempo.

A estrutura de dados do arquivo que armazena informações de tempo, *MRO_tempo*, é formada por um "vetor" de componentes de tempo (*v_tempo*). Cada componente do "vetor" representa um atributo temporal no qual podem ser armazenados os valores de tempo. A estrutura de dados é a seguinte:

```
struct MROTempo { v_tempo  
tempos[MAX_TEMPO]; } MRO_tempo;
```

A variável *MAX_TEMPO* determina o número máximo de atributos temporais representados que atuam simultaneamente em um atributo temporizado. Nesta implementação optou-se por definir *MAX_TEMPO* igual a quatro. O número definido mostrou-se suficiente para as necessidades identificadas. No capítulo oito apresentam-se considerações a este respeito.

A figura 5.7 apresenta o caminho que determina os valores de um atributo temporizado. Nessa figura utilizam-se apenas dois atributos temporais, o que implica na utilização dos dois primeiros componentes do registro de tempo, *MRO_tempo*.

7.2.5 - Estrutura de Dados do Ambiente.

Na seção 6.7 descreveram-se as variáveis necessárias para o estabelecimento do ambiente, que possibilitam a manipulação dos valores de tempo a serem utilizados nas operações da BD. A estrutura de dados definida para essas variáveis, Real e Assumido, é a seguinte:

```
struct { pal2 granularidade;  
  v_tempo real, assumido;  
  } MRO_amb tempo[MAX_TMP_USR+2];
```

```
struct MROTripla {  
  coditipo tipo;  
  coditipo subtipo;  
  codirel prt[4];  
  coditmp tempovida;  
  codiatr atrib;  
} MRO_tripla;
```

Figura 7.4 - Arquivo de Relacionamento Triplo

```
struct MROPropriedade {  
  coditipo tipo;  
  codipro prox_pro,  
           volta_pro;  
  union B_valores valor;  
  coditmp reg_tempo;  
} MRO_propriedade;
```

Figura 7.5 - Arquivo de Propriedade

Como apresentado na figura 6.1, cada atributo temporal (representado por cada componente de *MRO_amb_tempo*) possui dois valores de tempo definidos, que são Real e Assumido, que possuem os limites inferior e o superior. A variável *MAX_TMP_USR* representa o número máximo de definições que podem ser realizadas para o atributo temporal tempo definido pelo usuário para o ambiente em um esquema. A definição de *MAX_TMP_USR+2* é devido aos atributos temporais pré-definidos: tempo da transação e tempo válido.

A granularidade representa a granularidade associada a cada atributo temporal no ambiente. O GEO Temporal suporta a representação de algumas unidades padrões de tempo utilizadas nas atividades humanas, e seus inter-relacionamentos. Neste caso, o usuário escolhe alguma granularidade mínima para cada aplicação, baseando-se na menor unidade de tempo que seja necessária. Os dados temporais são armazenados em uma unidade padrão, que independe da granularidade escolhida pelo usuário. Existe uma tabela de conversão entre a unidade padrão e as unidades de tempo utilizadas nas atividades humanas. Deste modo, quando o usuário solicitar a apresentação de alguma informação de tempo, haverá a conversão da unidade padrão para a granularidade escolhida pelo usuário.

7.3 - Rotinas Básicas de Acesso a Tempo.

Para a implementação dos conceitos temporais desenvolvidos neste trabalho, realizou-se a especificação e a construção de rotinas básicas que possibilitassem a manipulação de atributos temporais e atributos temporizados pelo Gerenciador. Este conjunto de rotinas básicas foram acrescentadas às já existentes no GEO para o tratamento dos demais elementos do MRO*.

Essas rotinas foram implementadas em linguagem "C" e estão agrupadas em rotinas para o esquema e rotinas para a BD. Elas podem ser classificadas segundo as operações que realizam tais como inserção, eliminação, consulta e atualização de tempo. No apêndice B apresenta-se um manual destas rotinas.

A denominação das rotinas segue um padrão pré-estabelecido para facilitar seu entendimento e memorização. Na definição dos nomes, as seguintes normas foram utilizadas:

- as três primeiras letras indicam qual a principal informação que a rotina devolve;
- as demais letras indicam o Tipo de elemento a ser consultado para a obtenção das informações desejadas.

Nas seções seguintes apresentam-se as rotinas segundo o seu agrupamento e suas finalidades.

7.3.1 - Rotinas de Acesso ao Esquema.

As rotinas desta classe são utilizadas para consultar as informações contidas no Esquema de Dados.

Essas rotinas são utilizadas para informar quais os atributos temporais que temporizam uma determinada propriedade, e para um atributo temporal específico associado a uma propriedade, qual é a sua granularidade, se é intervalo ou evento, e a densidade.

O suporte a atributos com características temporais pelo GEO foi realizado construindo-se primitivas de manipulação de atributos temporais que seguiram o mesmo padrão adotado na construção de outras características de atributos já implementadas. Tal padrão envolve fundamentalmente primitivas para:

- a) Posicionar atributos temporais para recuperação no Esquema;
- b) Obter atributos temporais no Esquema.

As rotinas são as seguintes:

1) TmpPRO (coditipo ctp, coditmp ctt1, coditmp ctt2, coditmp ctt3, coditmp ctt4)

Descrição: Dado o código do tipo de propriedade retorna os códigos de tipo de tempo que afetam essa propriedade.

2) DadTMP (coditipo ctp, coditipo ctt, pal2 granu, pal2 interv, pal2 densidade)

Descrição: Dados os códigos dos tipos de propriedade e do atributo temporal, retorna a granularidade, se é intervalo ou evento, e a densidade.

Note-se que o Esquema de uma aplicação é armazenado na base como uma colônia, portanto pode ser manipulado e consultado usando os recursos usuais de manipulação e consulta da base. As primitivas de consulta ao esquema são especializações de primitivas de consulta à base, construídas como uma facilidade adicional àquelas.

7.3.2 - Rotinas de Acesso a Base de Dados.

As rotinas desta classe são utilizadas para a manipulação de valores de tempo de atributos temporais associados a atributos temporizados, armazenados na BD.

O suporte a atributos com características temporais pelo GEO foi realizado construindo-se primitivas de manipulação de valores de tempo que seguiram o mesmo padrão adotado na construção de outras características de atributos já implementadas. Tal padrão envolve fundamentalmente primitivas para:

- a) Colocar atributos temporizados na BD;
- b) "Retirar" atributos temporizados da BD;
- c) Posicionar atributos temporizados para recuperação na BD;
- d) Obter atributos temporizados da BD;
- e) Estabelecer o estado de acesso de como os atributos temporais determinam a manipulação dos atributos temporizados.

As rotinas são as seguintes:

1) PNPROTMP (union B_valores valor, coditipo CTA, v_tempo vt1, v_tempo vt2, v_tempo vt3, v_tempo vt4)

Descrição: Posiciona na propriedade do tipo CTA com o valor indicado e os valores de tempo fornecidos do Objeto/Relacionamento corrente.

2) CTTMP (v_tempo vt1, v_tempo vt2, v_tempo vt3, v_tempo vt4)

Descrição: Esta rotina grava os valores de tempo para o Objeto/Relacionamento corrente.

3) ANTMP

Descrição: Inicializa o registro de valores de tempo.

4) BDTMP (codipro ctp, v_tempo vt1, v_tempo vt2, v_tempo vt3, v_tempo vt4)

Descrição: Lê os valores de tempo da propriedade corrente.

5) PNTMP

Descrição: Posiciona no registro de tempo da propriedade corrente.

6) TATMP

Descrição: Posiciona na primeira ocorrência do registro de tempo da propriedade corrente.

7) COMPTMP (v_tempo vt1, v_tempo vt2, v_tempo vt3, v_tempo vt4)

Descrição: Compara os valores de tempo da propriedade corrente com os valores passados na rotina.

7.4 - Impacto nas Rotinas de Propriedade.

Devido a decisão de neste trabalho temporizar-se somente a característica Propriedade do MRO*, apenas a estrutura de dados e as rotinas de propriedade foram impactadas pela temporização do MRO*/GEO. As alterações realizadas na estrutura de dados de Propriedade foram abordadas na seção 7.2.3. Quanto as rotinas, na seção 7.4.1 abordam-se as que são impactadas.

7.4.1 - Rotinas de Propriedades Impactadas.

São apresentadas a seguir as rotinas impactadas, junto com a descrição do que realizavam e o que devem realizar devido a temporização.

- BDPRO: Lê a propriedade corrente informando qual é o Código de Tipo de Atributo (CTA) e qual o seu valor (Valor). Se a propriedade for temporizada deve informar os valores de tempo.

- CTPRO: Coloca uma propriedade do tipo CTA com o valor indicado no Objeto/Relacionamento corrente. Se a propriedade for temporizada deve inserir os valores de tempo associados.

- RNPRO: Esta rotina elimina a propriedade corrente do Objeto/Relacionamento corrente. Se a propriedade for temporizada deve ser realizada a eliminação lógica do Objeto/Relacionamento.

A seguir apresenta-se o pseudo-código da rotina BDPRO na situação anterior e em seguida como ela fica realizando o tratamento à tempo.

Situação atual:

- 1- Verificar se o atributo está posicionado para Objeto ou Relacionamento.
- 2- Se não houver Objeto ou Relacionamento corrente enviar mensagem de erro.
- 3- Verificar se existe Propriedade corrente.
- 4- Se não houver Propriedade corrente enviar mensagem de erro.
- 5- Ler o tipo da Propriedade corrente.
- 6- Ler o valor da Propriedade.
- 7- Fim.

Situação com temporização:

- 1- Verificar se o atributo está posicionado para Objeto ou Relacionamento.
- 2- Se não houver Objeto ou Relacionamento corrente enviar mensagem de erro.
- 3- Verificar se existe Propriedade corrente.
- 4- Se não houver Propriedade corrente enviar mensagem de erro.
- 5- Verificar se existe Tempo corrente.
- 6- Se não houver Tempo corrente enviar mensagem de erro.
- 7- Ler o tipo da Propriedade corrente.
- 8- Ler o valor da Propriedade.
- 9- Ler os valores de tempo dos atributos temporais associados.
- 10- Fim.

Note-se que a interface a nível de programação de todas as rotinas de manipulação de Propriedade não foi alterada. Assim aplicativos que não consideram o uso de tempo não precisam ser alterados. Toda a manipulação do tempo é efetuada internamente, cuja parametrização é feita com as rotinas de manipulação de tempo. Assim um aplicativo que foi escrito sem considerar aspectos temporais pode acessar informações temporizadas, "aceitando" o ambiente de tempo definido previamente à execução desse aplicativo.

7.5 - Considerações Finais.

Neste capítulo descreveu-se a implementação das características temporais incorporadas ao MRO*. Esta implementação foi realizada no Gerenciador de Objetos (GEO). A versão deste gerenciador e portanto as próprias primitivas deste trabalho, foram desenvolvidas na linguagem de programação "C", em microcomputadores da linha PC-XT, tendo como

sistema operacional o MS-DOS. Foram desenvolvidas nove rotinas para realizar o tratamento à tempo no GEO. O número de linhas codificadas é de aproximadamente 500.

No próximo capítulo são apresentadas as conclusões obtidas com a realização deste trabalho e sugestões de futuras pesquisas.

Capítulo 8

Conclusões e Sugestões de Futuras Pesquisas

8.1 - Considerações Iniciais.

Este capítulo tem por objetivo apresentar as principais conclusões acerca deste trabalho, apresentar algumas decisões tomadas durante a realização deste projeto que julgam-se decisivas para os resultados finais atingidos e sugerir a realização de futuras pesquisas que lhe deem continuidade.

8.2 - Decisões de Projeto.

8.2.1 - Estrutura de Dados.

A estrutura de dados definida inicialmente para suportar a temporização do MRO*/GEO previa um ponteiro para uma estrutura de tempo no registro de propriedades. Esta estrutura de tempo difere da solução adotada pois previa ponteiros para o registro de tempo seguinte e para o anterior associados a uma propriedade de um tipo com um determinado valor. Ou seja, uma propriedade de um tipo com um determinado valor poderia estar associada a vários registros de tempo. A figura 8.1 exemplifica esta situação.

A solução adotada inicialmente apresenta maior poderio na execução de alguns tipos de operações com tempo pela possibilidade de navegação na estrutura de tempo. Para operações do tipo: "em que instante um determinado valor foi/é/será válido ?" a solução inicial é melhor pois após o posicionamento no registro de propriedade com o valor pesquisado, a pesquisa para saber em que instante o valor foi/é/será válido é feita somente no(s) registro(s) de tempo que possui(em) ligação uns com os outros. Na solução final é necessário pesquisar o único registro de tempo para cada registro de propriedade que satisfaça o valor pesquisado para se conhecer o instante de tempo.

A decisão de alterar a estrutura para a que foi adotada neste trabalho foi em função da realização de vários outros tipos de operações para as quais a solução final é melhor. Por exemplo, para uma operação do tipo: "qual é o valor que uma propriedade possui hoje ?" a solução adotada é melhor pois é necessário pesquisar o(s) registro(s) de tempo para se obter o registro que satisfaça a condição "hoje". No caso da solução adotada existe no máximo uma ocorrência do registro de tempo e no caso da solução inicial podem existir vários registros, o que implicará em maior tempo de pesquisa.

Figura 8.1 - Representação da estrutura de dados adotada inicialmente.

8.2.2 - Propriedades Semânticas.

As propriedades semânticas granularidade, densidade e intervalo/evento foram associadas ao atributo temporal que se relaciona com os atributos temporizados através da característica propriedade do MRO* conforme figura 5.5.

A propriedade tempo de vida foi associada aos componentes do MRO* passíveis de tratamento temporal, no caso Objeto, Relacionamento Binário e Relacionamento Triplo.

8.3 - Sugestões de Futuras Pesquisas.

Com base neste trabalho sugere-se o desenvolvimento de novas pesquisas que investiguem alguns tópicos que são apresentados nesta seção.

8.3.1 - Colônias.

No MRO* não existe a associação de atributos com colônias e a incorporação de atributos a colônias tem consequências que não foram estudadas e fogem ao escopo deste trabalho. Sugere-se estudar a associação de atributos a colônias e após isto estudar a

associação de **atributos temporais aos atributos das colônias**.

Não foi realizada a análise do impacto da temporização de atributos ao gerenciamento de versões de colônias, embora tal análise mereça ser futuramente empreendida.

8.3.2 - Características do MRO*.

Como apresentado no capítulo 4 o MRO* possui características tais como sinônimo, comentário, estrutura, regra, propriedade. Neste trabalho realizou-se a temporização da característica propriedade. Todas as outras características de atributos do MRO* podem a princípio ser associadas a atributos temporais. No entanto a **temporização das outras características** requer estudos mais elaborados, tal como a temporização de comentários. Nesse caso percebe-se que a evolução de um texto através de várias edições sucessivas não pode ser tratado da mesma maneira como se trata por exemplo a substituição de um valor de uma propriedade por outro valor. Contudo, devem ser avaliadas as maneiras como partes de textos são sucessivamente alteradas. Este trabalho estabelece os critérios gerais de como tempo pode ser suportado no MRO* e não cabe aqui o detalhamento de como cada característica de atributo deve ser manipulada, uma vez temporizada. Assim utiliza-se a característica de propriedade para exemplificar como os critérios devem ser empregados e embora trate-se exaustivamente a temporização de propriedades, as demais características de atributos não são abordadas.

Em relação à característica regra especificamente, é necessário um estudo para especificar como tal característica é afetada pela temporização. O estudo do elemento temporal em conjunto com o conceito de regras pode levar à construção de uma BD onde a passagem do tempo provoca a realização de uma série de operações sem que o usuário tenha conhecimento. Ou seja, a associação de **tempo e regra** conduz as BDs Dinâmicas onde um evento temporal pode causar a execução de regras, que por sua vez podem provocar a realização de operações sem que o usuário tenha conhecimento. Bases de Dados Orientadas a Objetos são dinâmicas pois guardam o comportamento e regra é comportamento. Além disso, regras que são temporizadas e alteradas estabelecem: critérios de consistência, operações automáticas, etc. que mudam com o tempo. Nesse sentido deve-se estudar particularmente o caso de haver construções sintáticas na linguagem de declaração e de definição de regras que vinculem a avaliação de condições a características temporais, como por exemplo através de operadores QUANDO <tempo>, ANTES_DE <tempo>, etc.

8.3.3 - Linguagem de Manipulação dos Dados.

A LAMRO linguagem de acesso ao MRO* teve alguns comandos modificados e alguns novos definidos para tratar os aspectos temporais desenvolvidos porém não foram implementados. Em termos de implementação da linguagem este trabalho não objetiva a realização da codificação dos comandos uma vez que tal tarefa não é parte integrante do objetivo principal deste trabalho. No entanto está sendo empreendida essa implementação

através da orientação de alunos de graduação.

Sugere-se o desenvolvimento de uma **linguagem de consulta gráfica temporal** levando-se em consideração a linguagem TBE apresentada na seção 2.3.3.

8.3.4 - Evolução de Esquema.

Este trabalho e em particular a implementação descrita no capítulo 7 permitem que temporizem-se os atributos dos objetos e relacionamentos de um esquema. No entanto o significado disso não foi analisado embora reconheça-se que o impacto que isso traz sobre os conceitos de definição e evolução de esquemas precisam ser estudados, em particular vinculado ao conceito de gerenciamento de **versões de colônias** em geral.

A temporização do conceito de **Esquema Suplementar** apresentado em [CAMO92] com granularidades diversas (calendário, ordinal) conduz a resultados semelhantes aos obtidos com controle de versões uma vez que a temporização de versões permite a reativação de versões do esquema em tempos abstratos. Assim, sugere-se o desenvolvimento de pesquisas que estudem a temporização de Esquema Suplementar que devem conduzir a evolução de Esquema Suplementar.

8.3.5 - Manutenção de Bases de Dados.

Sugere-se o desenvolvimento de pesquisas que estudem as consequências da incorporação de tempo ao SGBD em tarefas de manutenção de BD tais como reestruturação, recuperação (recovery).

8.3.6 - Armazenamento de Dados.

Em implementações de BDTs deve-se observar que operações com dados não temporais não devem ser penalizadas, relativamente a rapidez e/ou lentidão de acesso aos dados. A separação dos dados armazenados em dados referentes ao estado **corrente e futuro**, e dados referentes ao **passado** é uma solução possível para este problema. É necessário também desenvolver técnicas para armazenamento e pesquisa de grandes quantidades de dados que devem ser armazenados e gerenciados. Na implementação do GEO Temporal realizada neste trabalho decidiu-se pela não separação devido a intenção de:

- a) o objetivo principal era mostrar a viabilidade de se realizar a implementação de BDTs;
- b) a partir da implementação realizada será possível a avaliação do que tem um desempenho satisfatório e o que não tem, refazendo-se o que não satisfizer;
- c) a carga de trabalho para realizar tal tarefa seria maior e mais complexa caso fosse realizada a separação. Quanto ao desempenho, a realização de tal separação induz a percepção de que ela deve ser melhor. Apenas com a implementação de tal separação pode-se obter subsídios para se proceder a tal avaliação. A estrutura de dados definida para

representar tempo permite que se realize tal separação.

8.3.7 - Granularidade.

Quanto a granularidade deve-se realizar estudos que viabilizem a realização de operações aritméticas tais como adição e subtração entre atributos temporais representados por granularidades distintas (calendário: abstrato, relativo; ordinal: abstrato) quando tais operações fizerem sentido.

8.4 - Considerações Finais.

Na seção 1.2 mencionou-se a existência de **dois desafios**. O **primeiro** é projetar modelos de dados e linguagens de consulta que permitam aos usuários projetar, implementar, consultar e atualizar BDTs. Esta tarefa foi realizada plenamente propiciando aos usuários os benefícios provenientes deste fato. O **segundo** é o desenvolvimento de técnicas para o armazenamento e pesquisa de grandes volumes de dados em BDTs. Esta tarefa foi parcialmente realizada, propondo-se e implementando-se uma extensão do GEO que suporte o modelo e a linguagem de consulta definida. Levando-se em consideração a experiência adquirida com a realização da implementação inicial obtêm-se subsídios para que se concretize o segundo desafio de forma eficiente.

No capítulo 3 foi apresentada a tabela 3.1 - sumário da análise dos modelos. Agora este trabalho é incluso na coluna G, possibilitando uma análise sintética do mesmo na tabela 8.1.

A representação de mais do que uma dimensão temporal num mesmo atributo é conceitualmente significativo e por isso decidiu-se pelo seu suporte na BD. No entanto, o significado genérico da associação de **dois ou mais atributos temporais** num mesmo atributo temporizado não foi empreendido. Este trabalho procura resolver o problema da representação e do suporte computacional desse fato, mas não é realizada uma análise conceitual do significado disso.

Nas experiências com a modelagem de dados temporais através do MRO^{*} temporal notou-se que os atributos temporais tempo da transação e tempo válido satisfizeram todas as necessidades surgidas. Segundo a bibliografia utilizada neste trabalho, particularmente [SNOD86a], [SNOD86b], e [SNOD87], propõem-se que o atributo temporal tempo definido pelo usuário deve ser representado devido a possíveis necessidades do usuário não atendidas pelos atributos temporais tempo da transação e tempo válido. Porém, essas necessidades não foram identificadas em nossa experiência visto que sempre conseguia-se utilizar tempo da transação e/ou tempo válido para satisfazer as necessidades surgidas. O tratamento ao atributo temporal tempo definido pelo usuário foi empreendido neste trabalho conduzindo a um elevado nível de complexidade. No entanto, embora teoricamente esse conceito pareça poderoso e necessário, uma vez dispondo do ferramental para utilizá-lo em situações práticas, não encontrou-se utilização para o mesmo. Assim, indica-se que a complexidade adicional oriunda desse suporte não se justifica diante da frequência com

que possa ser utilizado. A utilização de tempo definido pelo usuário generaliza o modelo temporal, sua linguagem de consulta, bem como sua implementação e por isso foi incorporado e suportado neste trabalho.

Deve-se ressaltar que a implementação desenvolvida neste trabalho é uma das poucas realizadas. Durante o desenvolvimento deste trabalho soube-se através da literatura especializada que somente o trabalho desenvolvido por Snodgrass em [SNOD87] foi implementado.

Observação : Na tabela 8.1 as linhas representam as características e as colunas os autores e os artigos.

- (1) Tipo de BD.
- (2) Modelo de dados básico.
- (3) Modelo conceitual temporal.
- (4) Desenvolvimento de um modelo para BDTs.
- (5) Linguagem de consulta temporal.
- (6) Atributo/objeto "time-stamp".
- (7) Ponto/intervalo de tempo.
- (8) Propriedades semânticas.
 - 8.1 Granularidade
 - 8.2 Tempo de vida
 - 8.3 Densidade
- (9) Implementação.
- (10) Aplicação alvo.

- (A) Segev e Shoshani ([SEGE87],[SEGE88]).
- (B) Snodgrass ([SNOD87]).
- (C) Tansel e outros ([TANS89a]).
- (D) Ariav ([ARIA86]).
- (E) Clifford e Warren ([CLIF83]).
- (F) Gadia ([GADI88]).
- (G) Silva.

Tabela 8.1 - Tabela 3.1 incluindo comparação com GEO Temporal

	A	B	C	D	E	F	G
1	histórica	tempo- ral	histó- rica	tempo- ral	histó- rica	histó- rica	temporal
2	TDM e relacional	rela- cional	rela- cional	rela- cional	rela- cional	rela- cional	orien- tado a objetos
3	sim	não	não	não	não	não	sim
4	sim	não	sim	sim	sim	não	sim
5	sim	sim	sim	sim	sim	sim	sim
6	atributo tupla	tupla	atributo	atributo	atributo	atributo	atributo
7	ponto	ponto e inter- valo	inter- valo	inter- valo	ponto	inter- valo	intervalo
8.1	sim	sim	sim	sim	sim	sim	sim
8.2	sim	sim	sim	sim	não	não	sim
8.3	sim	não	não	não	não	não	sim
9	não	sim	não	não	não	não	sim
10	científicas e estatísti- cas	?	?	sistemas de apoio a decisão	?	?	Projeto

Bibliografia

- [ALLE83] ALLEN, J.F. *Maintaining Knowledge About Temporal Intervals* Communication of the ACM, V.26, N.11, p.832-43, Nov. 1983.
- [AHN*86] AHN, I., SNODGRASS, R. *Performance Evaluation of a Temporal Database Management System* Sigmod Record, V.15, N.2, p.96-107. /Apresentado na *International Conference on Management of Data, Washington, 1986-Proceedings/*
- [AHN*88] AHN, I., SNODGRASS, R. *Partitioned Storage for Temporal Databases* Inf. Systems, V.13, N.4, p.369-391, 1988.
- [ARLA86] ARLAV, G. *A Temporally Oriented Data Model* ACM Transactions on Database Systems, V.11, N.4, p.499-527, Dec. 1986.
- [AUTO**] AUTOCAD REFERENCE MANUAL. Autodesk.
- [BLAJ92] BLAJIZ, M. *MRO* - Uma Atualização do Modelo de Representação de Objetos e Uma Abordagem Funcional*, Dissertação de Mestrado, ICMSC-USP, 1992.
- [BIC*86] BIC, L., GILBERT, J.P. *Learning from AI: New Trends in Database Technology* IEEE Computer, p.44-54, March 1986.
- [BOLO82] BOLOUR, A., et al. *The Role of Time in Information Processing: A Survey*. ACM SIGMOD Record, V.12, N.3, p.27-50, Apr. 1982.
- [BOLO83] BOLOUR, A., DEKEYSER, L.J. *Abstractions in Temporal Information* Inf. Systems, V.8, N.1, p.41-49, 1983.
- [CAMO92] CAMOLEZI, L. *Suporte a Acesso Multi-Usuário em Bases de Dados Orientadas a Objetos Através de Esquemas Suplementares*, Dissertação de Mestrado, ICMSC USP, 1992.
- [CENS89] CENSI, A., *AutoCAD 10, Guia Prático*, São Paulo, Érica, 1989.
- [CLIF83] CLIFFORD, J., WARREN, D. *Formal Semantics for Time in Databases* ACM Trans. on Database Systems V.8, N.2, p.214-265, June 1983.
- [Codd70] CODD, F. *A Relational Model for Large Shared Data Banks* Communications of ACM V.13, N.167, p.377-387, Junho 1970.
- [Codd79] CODD, F. *Extending the Relational Database Model to Capture More Meaning* ACM Transactions Database Systems V.4, N.4, p.397-434, Dec.1979.

- [COPE82] COPELAND, G. *What If Mass Storage Were Free ?* IEEE Computer p.27-35, July 1982.
- [DATE86] DATE, C.J. *Introdução a Sistemas de Bancos de Dados 4.ed.*, São Paulo, Campus, 1986.
- [DEAN85] DEAN, T.L., McDERMOTT, D.V. *Temporal Data Base Management* Artificial Intelligence, V.32, N.1, p.1-55, 1985.
- [ELMA91] ELMASRI, R. *Recent Advances In Temporal Databases* In : Simpósio Brasileiro de Banco de Dados, 6, 1991, Anais, Manaus, pg.17-26.
- [GADI88] GADIA, S.K. *A Homogeneous Relational Model and Query Languages for Temporal Databases* ACM Trans. on Database Systems, V.13, N.4, p.418-48, Dec. 1988.
- [LEE*85] LEE, R.M. *et all*, *Temporal Inferencing on Administrative DB.* Inf. Systems, V.10, N.2, p.197-206, 1985.
- [LUM*84] LUM, V., *et all* *Designing DBMS Support for the Temporal Dimension* SIGMOD Record, V.14, N.10, p.115-30, 1984. /Apresentado no SIGMOD'84 - Proceedings of Annual Meeting, Boston, 1984./
- [MCKE86] McKENZIE, E. *Bibliography: Temporal Databases* ACM SIGMOD Record, V.15, N.4, p.40-52, Dec. 1986.
- [MCKE87] McKENZIE, E., SNODGRASS, R. *Extending the Relacional Algebra to Support Transaction Time* In : Proc. ACM SIGMOD Int. Conf. on Management of Data, 1987, p.467-478.
- [OZSO84] OZSOYOGLU, Z.M., OZSOYOGLU, G. *STBE - A Database Query Language for Manipulating Summary Data* In : Proc. IEEE COMPDEC Conf., 1984.
- [PIZZ92] PIZZIGATTI, P.L.C. *Uma Linguagem Para o Gerenciador de Objetos Baseada no Modelo de Representação de Objetos*, - Dissertação de Mestrado, ICMSC-USP, 1992.
- [SEGE87] SEGEV, A., SHOSHANI, A. *Logical Modeling of Temporal Data* In : Proc. of the ACM SIGMOD Int. Conf. on Management of Data, May 1987, pg. 454-466.
- [SEGE88] SEGEV, A., SHOSHANI, A. *The Representation of a Temporal Data Model in the Relational Environment* In : International Working Conference SSDBM, 4., Roma, 1988 Proceedings. New York, Springer Verlag, 1989. (Lecture Notes in Computer Science, 339. ("Invited paper"))

- [SILV92] SILVA, R.C. *Armazenagem e Gerenciamento de Informações Temporais em um Modelo Orientado a Objetos* Simpósio Brasileiro de Banco de Dados, 7, Porto Alegre-RS, p.93-107, 1992. Anais.
- [SNOD86a] SNODGRASS, R., AHN, I. *Temporal Databases* IEEE Computer, p.35-42, Sept. 1986.
- [SNOD86b] SNODGRASS, R. *Research Concerning Time in Databases : Project Summaries* ACM SIGMOD Record, V.15, N.4, p.19-39, Dec. 1986.
- [SNOD87] SNODGRASS, R. *The Temporal Query Language TQuel* ACM Trans. on Database Systems, V.12, N.2, p.247-298, June 1987.
- [SNOD90] SNODGRASS, R. *Temporal Databases: Status and Research Directions* ACM SIGMOD Record, V.19, N.4, p.83-89, Dec. 1990.
- [SOO*91] SOO, M.D. *Bibliography on Temporal Databases* ACM SIGMOD Record, V.20, N.1, p.14-23, March 1991.
- [STAM88] STAM, R., SNODGRASS, R. *A Bibliography on Temporal Databases* IEEE Database Engineering, V.7, N.4, p.231-239, Dec. 1988.
- [STON76] STONEBRAKER, M., et al *The Design and Implementation of INGRES* ACM Trans. on Database Systems, Sept. 1976.
- [STON86] STONEBRAKER, M., ROWE, L.A. *The Design of Postgres* In : Proc. of the ACM SIGMOD Int. Conf. on Management of Data 1986, p.340-355.
- [TANS86] TANSEL, A.U. *Adding Time Dimension To Relational Model and Extending Relational Algebra* Inf. Systems, V.11, N.4, p.343-355, 1986.
- [TANS89a] TANSEL, A.U., ARKUN, M.E. *Time-by-Example Query Language for Historical Databases* IEEE Trans. on Soft. Engineering, V.15, N.4, p.464-478, April 1989.
- [TANS89b] TANSEL, A.U., GARNETT, L. *Nested Historical Relations* In : Proc. of the ACM SIGMOD Int. Conf. on Management of Data 1989, p.284-293.
- [TRAI86] TRAINA, C. *Máquina e Modelo de Dados Dedicados Para Aplicações de Engenharia*, Tese de Doutorado, IFQSC-USP, 1986.
- [TRAI88] TRAINA, C., SLAETS, J.F. *Um Modelo de Representação de Objetos* In : Simpósio Brasileiro de Banco de Dados, 3, Recife, 1988. Anais, p.227-242.
- [TRAI91] TRAINA, C. *GEO-Um Sistema de Gerenciamento de Bases de Dados Orientado*

a Objetos: Estado Atual de Desenvolvimento e Implementação In : Simpósio Brasileiro de Banco de Dados, 6, Manaus, 1991. Anais, p.193-207.

[ZLOO77] ZLOOF, M.M. *Query-by-Example : A Database Language* IBM System Journal, V.16, N.4, 1977.

[WIED86] WIEDERHOLD, G. *Views, Objects, and Databases* IEEE Computer, p.37-44, Dec. 1986.

APÊNDICE A

Exemplo de Utilização da LAMRO

Este apêndice mostra a utilização dos comandos da LAMRO de definição do esquema da aplicação de informações financeiras apresentado no capítulo quatro e utilizado neste trabalho.

/* Esquema da Aplicação de Informações Financeiras */

DEFINE OBJETO Aplicação, Indicador_Econômico
HABITA GLOBAL;

DEFINE OBJETO Informação_Financeira, Unidade_Organizacional
HABITA Aplicação;

DEFINE RELACIONAMENTO Possui
OPOSTO É_possuída
ORIGEM Unidade_Organizacional
DESTINO Informação_Financeira
MÍNIMO 1 MÁXIMO N
MÍNIMO OPOSTO 1
MÁXIMO OPOSTO N;

DEFINE RELACIONAMENTO Expressa
OPOSTO É_expressa
ORIGEM Indicador_Econômico
DESTINO Informação_Financeira
MÍNIMO 1 MÁXIMO N
MÍNIMO OPOSTO 1
MÁXIMO OPOSTO 1;

DEFINE PROPRIEDADE Nome
TIPO DE DADO Cadeia
ASSOCIADO OBJETO Informação_Financeira
OBJETO Unidade_Organizacional
OBJETO Indicador_Econômico;

DEFINE PROPRIEDADE Tipo
TIPO DE DADO Cadeia
ASSOCIADO OBJETO Informação_Financeira;

DEFINE PROPRIEDADE Classificação
TIPO DE DADO Cadeia

ASSOCIADO OBJETO Informação_Financeira;

DEFINE PROPRIEDADE Tipo_Uniorg
 TIPO DE DADO Cadeia
 ASSOCIADO OBJETO Unidade_Organizacional;

DEFINE PROPRIEDADE Valor_Informação_Financeira
 TIPO DE DADO Valor
 AFETADO POR Tempo_Válido, Tempo_Transação
 ASSOCIADO RELACIONAMENTO Expressa;

DEFINE PROPRIEDADE Total
 TIPO DE DADO Real
 AFETADO POR Tempo_Válido, Tempo_Transação
 ASSOCIADO OBJETO Informação_Financeira;

DEFINE PROPRIEDADE Sigla
 TIPO DE DADO Cadeia
 ASSOCIADO OBJETO Indicador_Econômico;

DEFINE PROPRIEDADE Valorização_Indicador
 TIPO DE DADO Valor
 AFETADO POR Tempo_Válido, Tempo_Transação
 GRANULARIDADE Dia
 DENSIDADE Passo_Constante
 INTERVALO
 ASSOCIADO OBJETO Indicador_Econômico;

/* Base de Dados da Aplicação de Informações Financeiras */

OBJETO Unidade_Organizacional São_Carlos;

OBJETO Informação_Financeira Depósito_Vista;

OBJETO Informação_Financeira Depósito_Prazo;

OBJETO Indicador_Econômico Dólar;

LOCALIZA OBJETO Indicador_Econômico Dólar;

ATRIBUTO Valorização_Indicador 21500
 Tempo_Válido EM '05.01.93'
 Tempo_Transação EM '05.01.93';

ATRIBUTO Sigla US\$;

LOCALIZA OBJETO Informação Financeira Depósito_Vista;
ATRIBUTO Classificação Captação;
ATRIBUTO Tipo Consolidado;
ATRIBUTO Total 4760000,00
Tempo_Válido EM '15.11.93'
Tempo_Transação EM '15.11.93';
ATRIBUTO Total 4873000,00
Tempo_Válido EM '17.11.93'
Tempo_Transação EM '17.11.93';
ATRIBUTO Total 5010100,92
Tempo_Válido EM '19.11.93'
Tempo_Transação EM '19.11.93';
ATRIBUTO Total 5320200,00
Tempo_Válido EM '20.11.93'
Tempo_Transação EM '20.11.93';

LOCALIZA OBJETO Informação Financeira Depósito_Prazo;
ATRIBUTO Total 12320000,00
Tempo_Válido EM '16.11.93'
Tempo_Transação EM '16.11.93';
ATRIBUTO Total 14300000,00
Tempo_Válido EM '17.11.93'
Tempo_Transação EM '17.11.93';
ATRIBUTO Total 18730000,00
Tempo_Válido EM '19.11.93'
Tempo_Transação EM '19.11.93';
ATRIBUTO Total 21350000,00
Tempo_Válido EM '20.11.93'
Tempo_Transação EM '20.11.93';

Dados da Aplicação de Informações Financeiras

Unidade Organizacional : São Carlos

Informação Financeira : Depósito à vista

Data do movimento : 15.11.92
Total : Cr\$ 4.760.000,00

Data do movimento : 17.11.92
Total : Cr\$ 4.873.000,00

Data do movimento : 19.11.92

Total : Cr\$ 5.010.100,92

Data do movimento : 20.11.92

Total : Cr\$ 5.320.200,00

Informação Financeira : Depósito à prazo

Data do movimento : 16.11.92

Total : Cr\$ 12.320.000,00

Data do movimento : 17.11.92

Total : Cr\$ 14.300.000,00

Data do movimento : 19.11.92

Total : Cr\$ 18.730.000,00

Data do movimento : 20.11.92

Total : Cr\$ 21.350.000,00

APÊNDICE B

Rotinas de Manipulação de Tempo

Manual de Referência

Considerações:

Este manual apresenta as rotinas que manipulam tempo em uma Base de Dados GEO. Tais rotinas podem ser utilizadas internamente por aplicativos ou diretamente por um programador.

NOME

PNPROTMP - Posiciona na propriedade, no valor, e nos valores de tempo do Objeto ou Relacionamento corrente.

SINOPSE

```
PNPROTMP(valor, CTA, vt1, vt2, vt3, vt4)
union B_valores valor;
coditipo CTA;
v_tempo vt1, vt2, vt3, vt4;
```

DESCRIÇÃO

Esta rotina posiciona na propriedade do tipo CTA com o valor indicado e os valores de tempo fornecidos do Objeto ou Relacionamento corrente.

RETORNO

.erro (vide ERROS).

ERROS

- 0 - não há erro.
- 403 - a propriedade indicada não é temporal.
- 404 - não existe registro de tempo para esta propriedade.
- 406 - não existe propriedade com esses valores de tempo indicados.

EXEMPLO

```
main()
{
    pal2 erro;
    coditipo CTA;
    union B_valores valor;
    v_tempo vt1, vt2, vt3, vt4;
    valor = _;
    CTA = 110;
    vt1.inicio = '05.01.1993';
    vt1.fim    = '06.01.1993';
    erro = PNPROTMP(valor, CTA, vt1, vt2, vt3, vt4);
    ...
}
```

NOME

CTTMP - Atribui valores de tempo ao atributo temporizado do Objeto ou Relacionamento.

SINOPSE

```
CTTMP(vt1, vt2, vt3, vt4)
v_tempo vt1, vt2, vt3, vt4;
```

DESCRIÇÃO

Esta rotina atribui os valores de tempo dos atributos temporais associados aos atributo temporizados do Objeto ou Relacionamento.

RETORNO

.erro (vide ERROS).

ERROS

- 0 - não há erro.
- 403 - a propriedade indicada não é temporal.
- 405 - já existe registro de tempo para a propriedade corrente.

EXEMPLO

```
main()
{
    pal2 erro;
    v_tempo vt1, vt2, vt3, vt4;
    vt1.inicio = '05.01.1993';
    vt1.fim    = '06.01.1993';
    erro = CTTMP(vt1, vt2, vt3,vt4);
    ...
}
```

NOME

ANTMP - Inicializa o registro de tempo.

SINOPSE

ANTMP()

DESCRIÇÃO

Esta rotina inicializa o registro de tempo MRO_tempo.

RETORNO

.erro (vide ERROS).

ERROS

0 - não há erro.

VEJA TAMBÉM

CTTMP(vt1, vt2, vt3, vt4);

EXEMPLO

```
main()
{
    pal2 erro;
    erro = ANTMP();
```


NOME

BDTMP - Informa os valores de tempo da propriedade corrente.

SINOPSE

```
BDTMP(ctp, vt1, vt2, vt3, vt4)
v_tempo vt1, vt2, vt3, vt4;
codipro ctp;
```

DESCRIÇÃO

Esta rotina realiza a leitura do registro de tempo MRO_tempo da propriedade corrente, informando os valores de tempo associados.

RETORNO

.erro (vide ERROS).

ERROS

- 0 - não há erro.
- 403 - a propriedade indicada não é temporal.
- 404 - não existe registro de tempo para esta propriedade.

EXEMPLO

```
main()
{
    pal2 erro;
    codipro ctp;
    v_tempo vt1, vt2, vt3, vt4;
    ctp = 107;
    vt1.inicio = '05.01.1993';
    vt1.fim    = '06.01.1993';
    erro = BDTMP(ctp, vt1, vt2, vt3,vt4);
```

NOME

PNTMP - Posiciona no registro de tempo da propriedade corrente.

SINOPSE

PNTMP()

DESCRIÇÃO

Esta rotina posiciona no registro de tempo da propriedade corrente.

RETORNO

.erro (vide ERROS).

ERROS

- 0 - não há erro.
- 9 - não existe propriedade corrente.
- 403 - a propriedade indicada não é temporal.
- 404 - não existe registro de tempo para esta propriedade.

VEJA TAMBÉM

TATMP()

EXEMPLO

```
main()
{
    pal2 erro;
    erro = PNTMP();
    ...
}
```

NOME

TATMP - Posiciona na primeira ocorrência do registro de tempo da propriedade corrente.

SINOPSE

TATMP()

DESCRIÇÃO

Esta rotina posiciona na primeira ocorrência do registro de tempo da propriedade corrente.

RETORNO

.erro (vide ERROS).

ERROS

- 0 - não há erro.
- 9 - não existe propriedade corrente.
- 403 - a propriedade indicada não é temporal.
- 404 - não existe registro de tempo para esta propriedade.

EXEMPLO

```
main()
{
    pal2 erro;
    erro = TATMP();
```

NOME

TMPPRO - Retorna os códigos de tipo de tempo que afetam uma propriedade.

SINOPSE

TMPPRO(ctp, ctt1, ctt2, ctt3, ctt4)

codipro ctp;

coditmp ctt1, ctt2, ctt3, ctt4;

DESCRIÇÃO

Esta rotina retorna os códigos de tipo de tempo que afetam uma propriedade, dado o código da propriedade.

RETORNO

.erro (vide ERROS).

ERROS

0 - não há erro.

402 - tipo de tempo não temporiza esse tipo de propriedade.

403 - a propriedade indicada não é temporal.

801 - código de tipo não encontrado.

805 - código especificado não é de um tipo de propriedade.

EXEMPLO

```
main()
```

```
{
```

```
    pal2 erro;
```

```
    codipro ctp;
```

```
    coditmp ctt1, ctt2, ctt3, ctt4;
```

```
    ctp = 1092;
```

```
    erro = TMPPRO(ctp, ctt1, ctt2, ctt3, ctt4);
```

NOME

DadTMP - Retorna as propriedades semânticas que afetam um atributo temporal.

SINOPSE

```
DadTMP(ctp, ctt, granu, interv, densidade)
codipro ctp;
coditmp ctt;
pal2 *granu, *interv, *densidade;
```

DESCRIÇÃO

Esta rotina retorna a granularidade, se é intervalo/evento, e a densidade de uma propriedade de um atributo temporal.

RETORNO

.erro (vide ERROS).

ERROS

- 0 - não há erro.
- 401 - código especificado não é de um tipo de tempo.
- 402 - tipo de tempo não temporiza esse tipo de propriedade.
- 801 - código de tipo não encontrado.
- 804 - código especificado não é de um tipo de propriedade.
- 812 - não existe o atributo especificado no relacionamento corrente.

EXEMPLO

```
main()
{
    pal2 erro;
    codipro ctp;
    coditmp ctt;
    pal2 *granu, *interv, *densidade;
    ctp = 1092;
    ctt = 3;
    erro = DadTMP(ctp, ctt, granu, interv, densidade);
    ...
}
```