

Regras de Consistências Temporais para Objetos e Relacionamentos Temporais

Solange Nice Alves de Souza

Faculdades Batista de Admin. e Informática
R. Prof. Homem de Mello, 537 – Perdizes
Cep: 05007-001 – São Paulo - SP
Brasil

solange@pcs.usp.br
snalves@cbb.g12.br

Edit Grassiani Lino de Campos

Depto. Eng. de Computação e Sistemas Digitais
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto - Travessa 3 – 158
Cidade Universitária – Butantã – São Paulo – SP
Cep: 005508-900 - Brasil
edit@pcs.usp.br

1. Resumo

Há um crescente número de aplicações que tem como requisito a manipulação da natureza temporal da informação.

A gerência do tempo envolve o controle de informações históricas, presentes e futuras ^[1]. Em modelos orientados a objetos ^[2, 3] com características temporais, os objetos, o contexto temporal a que pertencem (passado, presente, futuro), seus relacionamentos, particularmente, “part-of” e “is-a”, impõem restrições que afetam as operações de inserção, remoção e atualização.

Este trabalho apresenta os aspectos de relacionamentos do tipo “is-a” e “part-of” no âmbito de objetos temporais de um modelo de objetos estendido com estruturas de dados temporais para armazenamento de valores de atributos variantes no tempo.

Palavras-Chaves:

Banco de dados temporais; tempo de transação; tempo válido; relacionamentos temporais e hierarquias “is-a” e “part-of”; regras temporais; gerenciamento de dados históricos, presentes e futuros; operações de inserção, remoção e atualização.

2. Introdução

Este trabalho situa-se no contexto de um banco de dados bitemporal (manipulam tempos válido e de transação) ^[4, 5], cujo objetivo é explorar as facilidades e informações que a manipulação de tempos válido e de transação podem oferecer.

Essa proposta baseia-se na inspeção das finalidades de utilização de informações temporais para as aplicações. A gerência de tempo válido permite ao usuário ter como serviço oferecido pelo SGBDT, um completo gerenciamento da informação temporal (histórica e futura) sem que tenha que se preocupar em controlá-las nas aplicações. A gerência de tempo de transação permite ao SGBDT oferecer como serviço um controle completo de todas as atividades do sistema, além de oferecer um nível de segurança a mais: se for efetuada alguma operação de má fé, o ABD pode invalidar (ou remover) o novo estado do BD.

Neste trabalho, quatro marcas de tempo (“timestamps”) são utilizadas para o registro de tempo de transação e tempo válido, duas para cada classe tempo, tendo-se, respectivamente: **TTi - tempo de transação inicial**: momento em que o objeto temporal foi inserido no banco de dados; **TTf - tempo de transação final**: momento em que o objeto temporal passa a pertencer ao contexto temporal passado, ou seja, torna-se histórico no contexto do SGBDT; **TVi - tempo válido inicial**: momento em que o objeto temporal inicia sua existência no contexto da aplicação; **TVf - tempo válido final**: momento em que o objeto temporal torna-se histórico por determinação do usuário.

TTi e **TTf** são gerados pelo sistema e não são acessível ao usuário, a não ser para fins de auditoria. **TVi** e **TVf** são definidos pelo usuário.

Os pares $\langle \text{TTi}, \text{TTf} \rangle$ e $\langle \text{TVi}, \text{TVf} \rangle$ definem intervalos, definidos como fechados, que correspondem ao tempo de vida do objeto ^[6, 7, 8] no contexto do SGBDT e no contexto da aplicação, respectivamente.

A granularidade de tempo é a propriedade que define a unidade das marcas de tempo. Admite-se que tempo válido seja associado a diferentes unidades de tempo, ou seja, os objetos podem apresentar diferentes granularidades para tempo válido. Por outro lado, tempo de transação deve apresentar a menor unidade de tempo que permita registrar a diferença de tempo entre duas atualizações. Assim, a granularidade de tempo de transação deve corresponder a do “chronon” (menor unidade de tempo ou unidade atômica) e ser única para todos os objetos no banco de dados ^[9].

3. Modelo de Dados Bitemporal Orientado a Objetos

O modelo proposto segue o paradigma de orientação à objetos ^[2] quanto à estrutura. Neste trabalho, são discutidos apenas a inclusão de extensões relativas a aspectos temporais e a gerência dos mesmos.

Neste modelo, os objetos podem ser temporais ou não. Objetos não temporais são similares aos de modelos de objetos convencionais. No restante desse trabalho, enfoca-se os detalhes associados a objetos temporais. Quando necessário, são feitas observações sobre objetos não temporais.

3.1. Objetos Temporais

Objetos temporais distinguem-se de objetos convencionais pela incorporação de atributos temporais e de marcas de tempo associadas a tempos válido e de transação para controle da variação de valores no tempo. Um objeto temporal consiste de uma tupla da forma:

$\langle \text{TTi}, \text{TTf}, \text{TVi}, \text{TVf}, \text{id}, \{A_i\}, \{A_t\} \rangle$, onde:

TTi, TTf - marcas de tempo de transação inicial e final, respectivamente;

TVi, TVf - marcas de tempo válido inicial e final, respectivamente;

id - representa o conjunto de atributos invariantes no tempo declarados como integrantes do identificador do objeto;

{Ai} - conjunto de atributos definidos como invariantes no tempo (não integrantes do **id**);

{Av} - conjunto de atributos variantes no tempo que façam parte do objeto.

No momento em que um objeto é criado, as regras de dependência temporal que garantem a consistência temporal e regem as operações (inclusão, atualização e remoção) que alteram o estado de um objeto, juntamente com as informações sobre a granularidade das marcas de tempo são utilizados pelo SGBDT para atribuir valores a **TTi** e **TTf**, e para verificar os valores atribuídos pelo usuário para **TVi** e **TVf**. As regras de dependência temporal são apresentadas a seguir:

$$\mathbf{TTi} \leq \mathbf{TTf} \vee (\mathbf{TTi} \neq \text{null} \wedge \mathbf{TTf} = \text{null}); \quad (1)$$

$$\mathbf{TVi} \leq \mathbf{TVf} \vee (\mathbf{TVi} \neq \text{null} \wedge \mathbf{TVf} = \text{null}); \quad (2)$$

Os objetos temporais podem ser **históricos** (ou **passado**), **presentes** e **futuros**. Os valores das marcas de tempo associados ao objeto temporal definem a qual contexto ele pertence. Objetos não temporais pertencem somente ao contexto temporal presente. As regras a seguir definem o contexto dos objetos.¹

▪ **Objetos históricos:**

$$\left. \begin{array}{l} \mathbf{TVi} \leq \mathbf{TVf} < \text{now}; \\ \mathbf{TTf} \neq \text{null} \wedge \mathbf{TTi} \leq \mathbf{TTf} < \text{now}; \\ \mathbf{TTf} > \mathbf{TVf}; \end{array} \right\} \quad (3)$$

• **Objetos presentes** (objetos cuja validade é atual):

$$\left. \begin{array}{l} (\mathbf{TVi} < \text{now} < \mathbf{TVf}) \vee (\mathbf{TVi} < \text{now} \wedge \mathbf{TVf} = \text{null}); \\ \mathbf{TTi} < \text{now} \wedge \mathbf{TTf} = \text{null}; \end{array} \right\} \quad (4)$$

▪ **Objetos futuros:**

$$\left. \begin{array}{l} (\text{now} < \mathbf{TVi} \leq \mathbf{TVf}) \vee (\text{now} < \mathbf{TVi} \wedge \mathbf{TVf} = \text{null}); \\ \mathbf{TTi} < \text{now} \wedge \mathbf{TTf} = \text{null}. \end{array} \right\} \quad (5)$$

3.1.2. Atributos Variantes no Tempo

Atributos variantes no tempo são aqueles cujos valores variam no tempo e para os quais é mantido um registro temporal. Esses atributos são representados por estruturas especiais denominadas **atributo complexo temporal (ACT)**, que consistem de um conjunto de valores para um atributo complexo (atributo constituído por dois ou mais atributos). Um ACT consiste de um conjunto de n-tuplas da forma: **<tti, ttf, tvf, valor-atributo₁, valor-atributo₂, ..., valor-atributo_n>**, onde: **tti, ttf, tvf** - correspondem, respectivamente, a tempo de transação inicial e final, tempo válido inicial e final dos **valor-atributo₁, valor-atributo₂, ..., valor-atributo_n**;

Similarmente à objetos, as marcas de tempo de transação e de tempo válido associadas ao ACT delimitam os intervalos, fechados, correspondentes ao tempo de ‘vida’ do **valor-atributo** no contexto, respectivamente, do SGBDT e da aplicação.

As regras básicas que regem os valores permitidos para as marcas de tempo associadas a um ACT, verificadas pelo SGBDT quanto à consistência temporal são as seguintes:

¹ Utilizam-se dois valores especiais: “**now**” (representa o momento atual), e “**null**” (representa um valor desconhecido). A atribuição explícita de “**null**” ou a não atribuição de um valor específico para atributos definidos como não obrigatórios ou para as marcas de tempo válido, é interpretado como um valor desconhecido, similar a sistemas convencionais.

$$tti \leq ttf \vee (tti \neq \text{null} \wedge ttf = \text{null}); \quad (6)$$

$$tvi \leq tvf \vee (tvi \neq \text{null} \wedge tvf = \text{null}); \quad (7)$$

As regras 8 e 9, a seguir, definem os valores permitidos para as marcas de tempo do ACT com relação as marcas de tempo do objeto temporal. Estas regras exprimem que o intervalo definido pelas marcas de tempo de ACTs deve estar contido no intervalo definido pelas marcas de tempo do objeto associado.

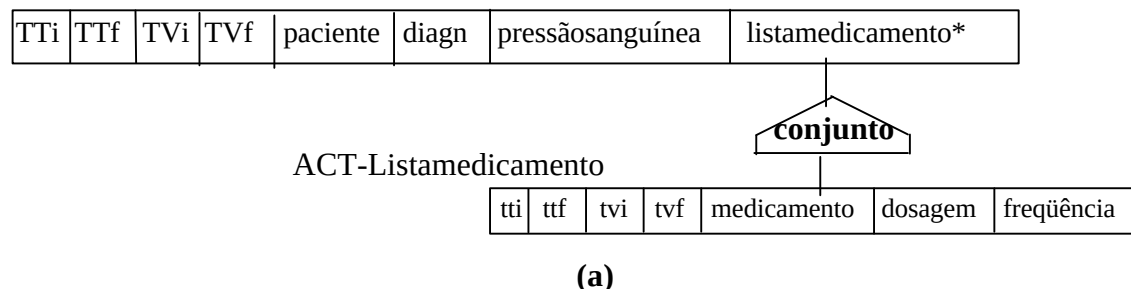
$$TTi \leq tti \wedge ttf \leq TTf; \quad (8)$$

$$TVi \leq tvi \wedge tvf \leq TVf; \quad (9)$$

A granularidade das marcas de tempo dos atributos é a mesma que aquela definida para o objeto correspondente.

Como exemplo, a figura 1.(a) mostra um esquema PACIENTE, que representa os problemas apresentados por um paciente durante uma consulta. Este, apresenta um atributo variante no tempo (com informações sobre os medicamentos prescritos para o paciente): *listamedicamento*², representado por um conjunto de n-tuplas da forma: **<tti,ttf,tvi,tvf,valor-medicamento,valor-dosagem,valor-freqüência>**.

PROBLEMA



PROBLEMA

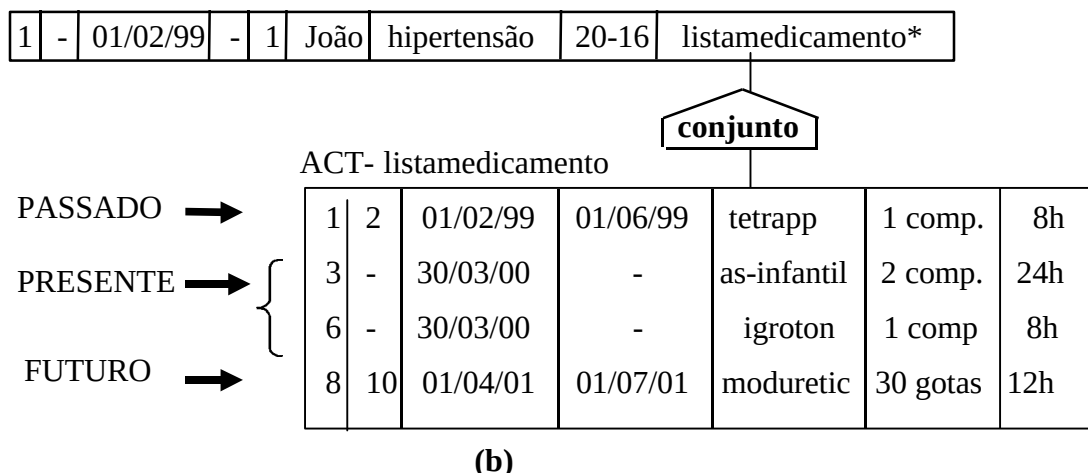


Fig. 1 - Representação do objeto temporal PACIENTE: (a) – Esquema conceitual com atributo ACT; (b) - Objeto PACIENTE com um conjunto de valores do ACT.

² Para efeito de representação, todos os atributos variantes no tempo têm associado ao seu nome o símbolo “*” para diferenciá-los dos invariantes no tempo.

A figura 1.(b) mostra o exemplo de um objeto PACIENTE com um conjunto de valores para o atributo *listamedicamento**. Nesta, a granularidade das marcas de tempo válido é dia/mês/ano; os valores apresentados para as marcas de tempo de transação são simbólicos (representando o valor do relógio do sistema). Observa-se, ainda na figura 1.(b) que:

- no contexto do SGBDT, um paciente, cujo diagnóstico é hipertensão, foi inserido no BDT, em **TTi** = 1 e permanece como objeto presente até que passe a ser um objeto histórico, o que ocorre quando expira seu tempo de validade, momento este no qual é inserido o valor “now” para **TTf**.
- no contexto da aplicação, o paciente João começou a apresentar hipertensão em 01/02/99 que ainda permanece (**TVf** = “null” no momento atual - “now”). A esse paciente foi receitado o medicamento tetrapp no período de [01/02/99, 01/06/99] de tempo válido, sendo esse valor registrado no BD no instante 1 de tempo de transação. Atualmente, João toma os medicamentos as-infantil e igroton (desde 30/03/00). Está prescrito para João o medicamento moduretic, para ser administrado por três meses, a partir de 01/04/01.

3.1.3. Contigüidade de Valores de Atributos

No modelo, os valores dos atributos ACT podem ser contíguos ou não contíguos. No caso de contíguos, os intervalos formados pelos valores registrados em **tv_i** e **tv_f** formam uma seqüência contínua, sem sobreposição de intervalos de tempo ou interrupção no tempo. No caso de não contíguos, essa restrição não existe; um exemplo é o atributo *listamedicamento**: foi prescrito mais de um medicamento para o paciente em instantes de tempo válido comuns (sobreposição dos intervalos de tempo entre dois elementos), ou ainda, instantes de tempo entre dois elementos em que não há medicamentos associados (descontinuidade entre os intervalos [**tv_i**, **tv_f**] de dois elementos). Estas duas categorias de atributos temporais são definidas para dar maior flexibilidade ao modelo, permitindo sua adequação às diferentes aplicações.

3.2. Relacionamentos entre Objetos

Relacionamentos não temporais existem apenas no contexto temporal presente e, quando removidos, desaparecem do BDT. As regras e tratamento são os mesmos de modelos convencionais, não sendo portanto, tratados aqui.

3.2.1. Relacionamentos Temporais

A exemplo das classes temporais, relacionamentos temporais também possuem marcas de tempo válido e de transação. As restrições de tempo impostas pelos objetos envolvidos no relacionamento impõem uma regra de integridade temporal sobre este relacionamento. Esta regra, controlada pelo SGBDT, é uma extensão direta do conceito de integridade referencial^[10, 11] e define que o relacionamento deve ser definido sobre instantes de tempo válidos comuns (sobreposição dos intervalos de tempo) aos objetos associados. Não teria sentido um relacionamento temporal entre dois objetos temporais disjuntos. Supondo que o objeto O₁ está associado ao objeto O₂ pelo relacionamento R, estas regras são definidas a seguir:

$$\begin{aligned} & \mathbf{TVi}^1 \leq \mathbf{TVi}^R \wedge (\mathbf{TVf}^R \leq \mathbf{TVf}^1 \vee \mathbf{TVf}^R = \mathbf{TVf}^1 = \text{null} \vee (\mathbf{TVf}^R \neq \text{null} \wedge \mathbf{TVf}^1 = \text{null})) \\ & \wedge \mathbf{TVi}^2 \leq \mathbf{TVi}^R \wedge (\mathbf{TVf}^R \leq \mathbf{TVf}^2 \vee \mathbf{TVf}^R = \mathbf{TVf}^2 = \text{null} \vee (\mathbf{TVf}^R \neq \text{null} \wedge \mathbf{TVf}^2 = \text{null}))^3 \quad (10) \\ & \text{sendo: } \mathbf{TVi}^R \leq \mathbf{TVf}^R \end{aligned}$$

Os valores registrados nas marcas de tempo válido definem o contexto temporal a que pertence o relacionamento. Este contexto pode ou não coincidir com a dos objetos relacionados, por exemplo,

³ Os índices 1, 2 e R são utilizados para indicar, respectivamente, marcas de tempo dos objeto O₁, O₂ e do relacionamento.

dois objetos presentes podem estar associados por um relacionamento passado, presente ou futuro, desde que a regra 10 seja obedecida.

Relacionamentos entre classes temporais podem ser temporais ou não temporais. Para que dois objetos temporais sejam associados por um relacionamento temporal, a regra 10 deve ser obedecida. Se o relacionamento é não temporal, então os objetos envolvidos devem pertencer ao contexto temporal presente, já que tal relacionamento só existe neste contexto.

Relacionamentos temporais podem também apresentar atributos variantes no tempo, valendo para estas regras e tratamentos similares aos definidos anteriormente para classes.

Todos os comentários feitos aplicam-se a relacionamentos de forma geral. No entanto, relacionamentos do tipo “**is-a**” e “**part-of**”^[2, 3], que definem hierarquias especiais entre objetos, serão considerados a seguir.

3.2.2. Hierarquia “IS-A” entre Objetos

Na estrutura hierárquica do tipo “**is-a**”, caso o usuário não atribua valores para as marcas de tempo válido de objetos inseridos nas subclasses, o SGBDT assume a propriedade da herança, e atribui os valores definidos para **TVi** e **TVf** dos objetos correspondentes na superclasse. Caso queira-se especificar valores diferentes daqueles para as subclasses, valem as regras e tratamentos gerais para relacionamentos entre objetos (regra 10).

Numa hierarquia “**is-a**”, devido à propriedade da herança, as subclasses devem ser compatíveis com a superclasse. Assim, superclasse temporal implica em subclasses temporais, e superclasse não temporal, em subclasses não temporais.

3.2.3. Hierarquia “PART-OF” entre Objetos

Em objetos complexos construídos a partir de relacionamentos do tipo “**part-of**” os objetos da superclasse e subclasse podem estar **fracamente acoplados** ou **fortemente acoplados**^[3].

Objetos Fracamente Acoplados

Em objetos fracamente acoplados não há dependência existencial, a remoção de um objeto da superclasse não implica na remoção do objeto componente da subclasse. A regra para relacionamentos entre objetos é aplicável a esta hierarquia (regra 10).

Objetos Fortemente Acoplados

Em relacionamentos “**part-of**” com objetos fortemente acoplados há dependência existencial de objetos da subclasse com relação aos da superclasse.

A regra para relacionamentos entre objetos, aplica-se a esta hierarquia, com algumas considerações:

Superclasse e subclasse não temporais: é o caso de dependência existencial em BDs convencionais, onde a remoção de um objeto da superclasse implica na remoção do objeto da subclasse.

Superclasse e subclasse temporais: devido a dependência existencial, o objeto da subclasse inicia e encerra sua validade dentro do intervalo de validade da superclasse $(TVi^{sb} \geq TVi^{sp} \wedge TVf^{sb} \leq TVf^{sp})^4$ (11)

4. Regras de Dependência Temporal

As operações de inserção, atualização e remoção sobre objetos temporais seguem regras de dependência temporal, que atuam sobre as marcas de tempo dos objetos e seus ACTs.

Essas regras têm como parâmetros básicos as **marcas de tempo válido** e **de transação** dos objetos e seus atributos temporais (**A_t**). Dependendo dos valores destes parâmetros, uma determinada

⁴ os índices ‘**sb**’ e ‘**sp**’ são utilizados para indicarem marcas de tempo do objeto, respectivamente, da subclasse e da superclasse.

operação poderá ser efetuada ou não. Além disso, tendo em vista as particularidades da gerência do tempo, novos operadores, **RemoveT** e **EraseT**, que estendem o conceito de remoção, são definidos. Estes e as extensões feitas aos operadores “insert”, “delete” e “update” convencionais, **InsertT**, **DeleteT**, **UpdateT**, respectivamente, são mostrados, a seguir.

4.1. Inserção de Objetos Temporais (InsertT)

Os parâmetros para este operador são: **InsertT** <classe, id, {**A_i**}, {**A_t**}, **TV_i**, **TV_f**>, onde:

classe - classe na qual o objeto está sendo inserido;

id - identificador do objeto;

{**A_i**} - conjunto de valores de atributos invariantes no tempo;

{**A_t**} - conjunto de valores de atributos temporais. Para cada **A_t**: **tvi**, **tvf**, **valor-atributo₁**, **valor-atributo₂**, ..., **valor-atributo_n** (que representam as marcas de tempo válido e os respectivos **valor-atributo** do atributo composto).

TV_i, **TV_f** - marcas de tempo válido.

Todos os objetos, sejam históricos, presentes ou futuros, são inseridos, desde que sejam obedecidas as regras de 1 a 10 definidas anteriormente.

É importante ressaltar a possibilidade de inserção de objetos instantâneos. Estes, possuem valores de **TV_i** e **TV_f** iguais, por força da granularidade de tempo da classe à qual pertencem. É o caso, por exemplo, em que um paciente é internado no hospital num certo dia e recebe alta no mesmo dia. Existem duas situações de inserção de um objeto instantâneo que merecem ser observadas: uma em que o usuário atribui valores iguais e diferentes de “null” para **TV_i** e **TV_f**; outra em que o usuário atribui valor de “now” a **TV_f** e “null” (valor desconhecido) a **TV_i**. Neste último caso, o objeto seria inserido com **TV_i** com valor “now”, o que, dependendo da granularidade de tempo válido da classe, representaria **TV_i** = **TV_f**, porém o usuário poderia ter cometido um erro na especificação do objeto, então nesta situação, a inserção é recusada. Diferentemente do primeiro caso, no qual o usuário definiu explicitamente valores iguais para **TV_i** e **TV_f**, onde a inserção é aceita.

4.2. Remoção de Objetos Temporais

De maneira geral, a maior parte dos trabalhos na área de banco de dados temporais ^[13, 12, 13, 14, 15] assumem que as informações, uma vez inseridas no BDT, não são mais removidas. Entretanto sabe-se que em muitos ambientes, as informações são de interesse até um certo momento, a partir do qual, realmente tornam-se obsoletas. Assim, a manutenção eterna de tais informações conduziria ao desperdício de espaço de armazenamento. Com essa visão, definem-se os operadores “**removeT**”, para remoção de conjuntos de objetos históricos, “**eraseT**” para remoção de objetos específicos, sejam históricos, presentes ou futuros e estende-se o operador “**deleteT**”, para transformação de objetos presentes em históricos. Esses operadores são discutidos a seguir.

4.2.1. Operador RemoveT

Os parâmetros deste operador são os seguintes: **RemoveT** <classe, [**Ti**, **Tf**]>, onde:

classe - classe sobre a qual a operação **removeT** será aplicada;

[**Ti**, **Tf**]⁵ - intervalo de tempo;

Esta operação pode ser utilizada para liberação do espaço de armazenamento utilizado por informações históricas que se tornaram obsoletas. O operador “**removeT**” atua sobre as marcas de

⁵ Os colchetes associados a parâmetros de tempo indicam um intervalo fechado.

tempo de transação e **só tem efeito sobre objetos históricos**. Tal operador remove todos os objetos históricos da classe indicada, cujo intervalo **[TTi, TTf]** esteja contido no intervalo de tempo **[Ti, Tf]** especificado na operação **removeT**.

O operador “**removeT**” é definido sobre as marcas de tempo de transação, pois estas são as marcas de tempo voltadas para o contexto do SGBDT, e o valor registrado em **TTf** define se o objeto é histórico;

Se o intervalo **[TTi, TTf]** de um objeto histórico está compreendido no intervalo **[Ti, Tf]** especificado na operação **removeT**, pela regra 10, todos os relacionamentos existentes que envolvam esse objeto, também terão seus intervalos **[TTi, TTf]** contidos no intervalo **[Ti, Tf]**.

Na hierarquia “**is-a**” e “**part-of**” com **forte e fraco acoplamento**, a remoção do objeto na superclasse, resulta na remoção dos objetos em subclasses, assim como todos os relacionamentos envolvendo qualquer desses objetos.

4.2.2. Operador EraseT

Os parâmetros deste operador são os seguintes: **EraseT <classe, id>**, onde:

classe - classe a qual o objeto pertence;

id - identificador do objeto;

O operador **eraseT** remove fisicamente objetos específicos (com seus respectivos ACTs) do BDT. Eventualmente, cometemos erros ao registrar dados num BD. Nestes casos, é importante a utilização de algum mecanismo que permita a correção desses erros. A operação “**eraseT**” é sugerida para essas situações (entrada incorreta de dados). Este operador pode ser aplicado a objetos históricos, presentes ou futuros.

Estendendo o conceito de integridade referencial, a remoção de um objeto histórico, presente ou futuro se dá na inexistência de um relacionamento envolvendo-o. Em hierarquias “**is-a**”, a remoção do objeto da superclasse implica na remoção do objeto da subclasse. Em hierarquias “**part-of**” com **acoplamento forte**, em função da dependência existencial, objetos em subclasses são removidos com a remoção do objeto da superclasse. Em hierarquias “**part-of**” com **acoplamento fraco**, objetos em subclasses são removidos somente se não estiverem associados a nenhum objeto na superclasse (este ficaria sem seu objeto componente). Na remoção de objetos em superclasses, seus objetos associados nas subclasses permanecem no BDT, pois não existe a dependência existencial.

4.2.3. Operador Delete T

Os parâmetros deste operador são os seguintes: **DeleteT <classe, P>**, onde:

classe - classe do objeto a ser removido;

P – predicado (parâmetro opcional). Pode ser utilizado para indicar quais objetos devem ser arquivados ou removidos do BDT. Caso esta cláusula seja omitida, a operação **deleteT** é aplicada a todos os objetos da classe.

Em BDs convencionais o operador “**delete**” tem um significado bem conhecido: remoção física de objetos. Devido ao aspecto temporal, no entanto, a funcionalidade deste operador é estendida e dependendo do contexto temporal do objeto, esse operador pode produzir os seguintes efeitos:

- sobre objetos históricos: não produz efeito algum, pois objetos históricos, em princípio, não são mais removidos do BDT, exceto em alguns casos, nos quais usam-se os operadores “**erase**” ou “**remove**”;
- sobre objetos presentes: aqui reside a diferença de interpretação deste operador em comparação com BDs convencionais. Neste caso, significa o arquivamento do objeto, isto é, a transformação do objeto presente em histórico. Nesta operação, o tempo válido do objeto é encerrado, o valor

atualmente registrado em **TVf** do objeto é alterado para um valor de tempo menor que “now” e **TTf** recebe o valor de tempo correspondente ao do momento atual; o objeto então, torna-se histórico. Todos os elementos de ACTs sofrem o mesmo efeito. Aqueles, cujos valores registrados em **tvf** e **ttf** apresentarem **tvf** < now e **ttf** = null, têm o valor de **ttf** modificado para “now”;

- sobre objetos futuros: remoção física do objeto. Objetos futuros são informações cuja validade temporal ainda não se iniciou, portanto não há porque guardar informações que não devem se realizar mais.

Objetos presentes podem estar associados a outros objetos por meio de relacionamentos históricos, presentes ou futuros, desde que a regra 10 seja válida. Quando um objeto é arquivado, o novo valor designado para seu **TVf** precisa ser checado com relação aos valores de **TVf^R** de relacionamentos existentes, pois o novo **TVf** pode implicar em objeto sem validade em instantes de tempo em que o relacionamento está definido (caso de novo **TVf** < **TVf^R**). Neste caso, considerando o conceito de integridade referencial é necessário alterar o valor de **TVf^R**, ou remover o relacionamento para arquivar o objeto. Como no modelo informações históricas não podem ser alteradas, a existência de relacionamentos históricos em que se tenha novo **TVf** < **TVf^R**, invalida a operação. Na existência de relacionamentos presentes em que novo **TVf** < **TVf^R** ou **TVf^R** = null, como não há restrições quanto a alteração de informações presentes, o valor de **TVf^R** é alterado para o valor designado para **TVf**. Se o relacionamento é futuro, não há por que ser mantido, o objeto é arquivado e o relacionamento é removido. Além disso, em hierarquias “is-a”, considerando as características dessa hierarquia, quando um objeto presente da superclasse é arquivado, seus objetos associados em subclasses também o são. Em hierarquias “part-of” com **acoplamento forte**, devido a dependência existencial, o arquivamento do objeto presente na superclasse implica no arquivamento dos seus objetos associados nas subclasses. Em hierarquias “part-of” com **acoplamento fraco**, no processo de arquivamento de objetos da superclasse, os objetos associados nas subclasses não são arquivados.

De forma similar, na aplicação de “deleteT” sobre objetos futuros em hierarquias “is-a” ou “part-of” com **acoplamento forte** ou **fraco** a existência de relacionamentos envolvendo algum dos objetos na hierarquia invalida a operação.

4.3. Atualização de Objetos Temporais (UpdateT)

A operação **updateT** quando aplicada a objetos temporais pode ser utilizada para alterar atributos invariantes ou variantes no tempo, ou marcas de tempo válido do objeto.

Os parâmetros deste operador são os seguintes:

UpdateT <classe, id, {**A_t**}, {**A_i**}, **TVi**, **TVf**, **P**> ,onde:

classe - classe a que pertence o objeto;

id - identificador do objeto;

{**A_t** } - representa o conjunto de atributos variantes no tempo. Para cada **A_t** (representado por um ACT): **tvi**, **tvf**, **valorA_{t1}**, ..., **valorA_{tm}** (que representam as marcas de tempo válido e os respectivos valores para os atributos que compõem **A_t**);

{**A_i** } - representa o conjunto de atributos invariantes no tempo.

TVi, **TVf** - marcas de tempo válido do objeto;

P - representa um predicado.

Com exceção da classe, todos os outros parâmetros são opcionais, e são fornecidos de acordo com o que se deseja alterar.

Em BDs convencionais, altera-se o valor de um atributo, fornecendo um novo, que passa a ser o seu valor atual. O valor anterior deste atributo é descartado, não podendo mais ser recuperado. No caso do modelo temporal aqui apresentado, quando o A_t é atualizado, o novo valor é inserido e o valor anterior não é descartado. A operação de atualização do A_t funciona como a inclusão de um elemento no **ACT**. Outra diferença em relação a BDs convencionais, é que o novo valor do atributo não necessariamente passa a ser seu valor atual. O usuário define, utilizando as marcas de tempo válido de elementos do **ACT**, a que contexto temporal o novo valor pertencerá.

A atualização de objetos temporais não se aplica a objetos históricos, pois no modelo assume-se que a história não é alterada. Assim, as discussões apresentadas a seguir com relação a atualização de objetos temporais, aplicam-se somente a objetos presentes e futuros.

A atualização de atributos invariantes no tempo (não integrantes do id) é similar a de BDs convencionais: uma vez alterado, o valor anterior do atributo é descartado, não podendo mais ser recuperado.

As regras para atualização de objetos temporais, visando atributos variantes no tempo e marcas de tempo válido dos objetos são discutidas a seguir.

Atributos Variantes no Tempo:

Na atualização de atributos variantes no tempo, pode-se atualizar os valores dos atributos (valor-atributo₁, ..., valor-atributo_n) registrados no elemento < **t_{ti}**, **t_{tf}**, **t_{vi}**, **t_{vf}**, **valor-atributo₁**, ..., **valor-atributo_n**> do **ACT**. Os valores registrados para as marcas de tempo válido do elemento são alterados pela atualização do **valor-atributo**.

Na atualização de um objeto, visando o atributo variante no tempo, dentre os demais parâmetros, o usuário especifica <**t_{vi}**, **t_{vf}**, **valor-atributo**>. Nesse sentido, valores para **t_{vi}** e/ou **t_{vf}** podem ter sido ou não fornecidos pelo usuário. Porém, similarmente a objetos, elementos de ACTs não são registrados com valor “null” para **t_{vi}**, neste caso, o valor atribuído a este pelo SGBDT dependerá da categoria, “contíguo” ou “não contíguo”, do ACT vinculado ao atributo variante no tempo, alvo da atualização. A atribuição de “null” a **t_{vf}** dependerá do valor de **TV_f** do objeto. Caso **TV_f** ≠ null, **t_{vf}** não poderá ter valor “null”, de forma que a regra 9 não seja infringida.

Marcas de Tempo Válido do Objeto

Similarmente a atributos invariantes no tempo, a operação de modificação aplicada às marcas de tempo válido funciona como a operação convencional: uma vez alterados os valores anteriores são descartados. De modo geral, na atualização de **TV_i** ou **TV_f**, a regra 9 não pode ser transgredida para manutenção da consistência temporal.

No caso de objetos presentes, **TV_i** pode ser menor ou igual a “now” (na medida que a granularidade de tempo da marca de tempo válido é mais grossa que a do “chronon”), sendo menor, é um valor histórico e, então, não sofre alteração; sendo igual, **TV_i** é alterado. Neste caso, se novo **TV_i** > now e existindo A_t : se **TV_i** > **t_{vf}** > **t_{vi}**, o elemento do ACT é removido (caso contrário, este seria válido em instantes de tempo em que o objeto não tem validade). Se **t_{vi}** < **TV_i** ≤ **t_{vf}**, **t_{vi}** é alterado recebendo o novo valor de **TV_i**. Se **TV_i** ≤ **t_{vi}**, a operação **updateT** é executada sem alteração no elemento do ACT. Procedimentos similares são seguidos para relacionamentos e elementos ACT vinculados a estes.

No caso de objetos futuros, se novo **TV_i** > **TV_i** atual (antecipação da validade do objeto), procedimentos similares ao discutido no caso anterior são seguidos na existência de A_t e relacionamentos.

No caso de alteração de **TV_f**, para objetos presentes ou futuros, se novo **TV_f** < **TV_f** atual e existindo A_t : se novo **TV_f** < **t_{vi}**, o elemento é removido. Se **t_{vi}** ≤ novo **TV_f** < **t_{vf}**, **t_{vf}** é alterado, recebendo o novo valor de **TV_f**. Se novo **TV_f** ≥ **t_{vf}**, a operação **updateT** é efetuada, sem necessidade

de alterações. Na existência de relacionamentos, procedimentos similares são acompanhados, sendo que se novo $TVf < TVf^R$ (ou $TVf^R = \text{null}$, sendo $TVf \neq \text{null}$) e $TVf < TVi^R$, a operação **updateT** é recusada.

5. Conclusões

Grande parte dos trabalhos na área de BDs temporais enfocam apenas o controle de informações históricas; em geral não existe preocupação com informações relativas ao futuro. Além disso, com algumas exceções^[4,12,13], apenas uma dimensão de tempo é gerenciada^[8, 16, 17]. Neste trabalho, houve a preocupação não só com a utilização dessas duas dimensões de tempo, mas principalmente com o levantamento da utilidade de cada uma para as aplicações.

O controle de duas dimensões de tempo (tempos válido e de transação), mais a existência de objetos históricos presentes e futuros aumentam a complexidade do modelo, sendo necessária a definição de regras temporais que tornem possível o gerenciamento correto de dados históricos, presentes e futuros. Essas regras devem considerar também o relacionamento entre essas dimensões de tempo. O objetivo principal deste trabalho foi o desenvolvimento de um modelo de dados temporal completo, a ser gerido por um SGBDT desonerando as aplicações desta tarefa. No entanto, mesmo que seja usado um SGBD convencional, as regras expostas seriam as mesmas.

Na implementação do modelo proposto todas as regras definidas neste trabalho devem ser consideradas, de modo a garantir a consistência temporal dos dados. Essas regras podem ser implementadas por uma camada (um “shell”) sobre um SGBD convencional, tirando proveito de facilidades que ele oferece, como, por exemplo, tipo de dados ‘data’, ‘hora e data’; funções de conversão de dados, a utilização de “triggers” e “stored-procedures” para a implementação de regras de dependência temporal e alterações do estado do BD, entre outras.

Snodgrass^[4] mostrou que o tempo de resposta a consultas que acessam dados atuais, que espera-se que sejam as mais freqüentes, sofre grande degradação quando informações presentes e históricas são armazenadas juntas. Assim, na implementação do modelo, é importante para o melhor desempenho do sistema, que dados históricos, atuais e futuros estejam fisicamente separados no meio de armazenamento.

Quando consultas são formuladas, cabe ao SGBDT identificar o contexto temporal a que o objeto pertence para fornecer as respostas corretas, sendo necessário que em algum momento, o SGBDT verifique os valores de TVi e TVf , comparando-os com o valor corrente de “now”. Este processo de verificação pode ser feito pela interrupção do relógio de tempo real. Neste caso, em certos intervalos de tempo, um processo interno ao SGBDT seria disparado e verificaria os valores de TVi e TVf de cada objeto, passando para o contexto histórico objetos presentes e para o contexto presente, objetos futuros. A desvantagem do processo de interrupção do relógio de tempo real é que o processo de verificação de TVi e TVf e atualização de Ttf e das marcas de tempo (ttf , tvi , tvf) de elementos de ACTs, quando objetos são passados para o contexto histórico, pode ser demorado e esse tempo tenderá a aumentar com o tamanho do BDT. Por causa desse procedimento, o tempo de resposta a consultas será mais alto que o habitual, e dependendo da freqüência com que o processo de interrupção é disparado, o desempenho do BDT sofrerá alta degradação. Uma alternativa ao processo de interrupção é a verificação do contexto temporal do objeto quando o mesmo é acessado. A cada consulta, o SGBDT primeiro verificaria os valores de TVi e TVf dos objetos envolvidos e, no caso de apresentar TVf menor que “now”, promoveria as atualizações necessárias, e responderia a consulta. Esta função de verificação poderia ser realizada em tempo de execução da consulta.

Referências Bibliográficas

- [1] Souza, Solange Nice Alves; Especificação de um Modelo de Dados Bitemporal Orientado a Objetos; Tese de Doutorado; Escola Politécnica; Universidade de São Paulo; 1998.
- [2] Catell, R. G. G.; Object Data Management – Object-Oriented and Extended Relational Database System; Addison-Wesley Company; 1994.
- [3] Dillon, Tharam; Tam, P. L.; Object-Oriented Conceptual Modeling; Prentice hall; 1993.
- [4] Snodgrass, R.; Temporal Database; Computer; v.19; n.9; september; pp.35-42;1986.
- [5] Jensen, C. S.; et al; A Glossary of Temporal Databases Concepts; Sigmod Record, v.21; n.3; september; 1992.
- [6] Roddick, J. F.; Patric, J. D.; Temporal Semantics in Information Systems – A Survey; Information Systems; v.17; n.3; pp.249-267; 1992.
- [7] Edelweiss, N.; Oliveira, J. P. M.; Modelagem de Aspectos Temporais de Sistemas de Informação; IX Escola de Computação; 24 a 31 de julho; Recife; 1994.
- [8] Blanken, H. Ijbema, A.; Language Concepts for Versioned Hierarchical Objects in Temporal Aspects in Information Systems; Rolland, C.; Bodart, F.; Leonard, M.; Elsevier Science Publishers B. V.; pp.187-201; 1998.
- [9] Jensen, C.S.; Snodgrass, R.; Semantics of Time-Varying Attributes and Their Use temporal Database Design; Technical Report R-95-2012; Departament of Mathematics and Computer Science; Alborg University; may; 1995.
- [10] Date, C. J.; Introdução a Sistemas de Banco de Dados; Editora Campos; 4ª edição.
- [11] Korth, H. F.; Silberschatz, A.; Sistemas de Banco de Dados; McGraw-Hill; 1989.
- [12] Edelweiss, N.; Oliveira, J. P. M.; Definição e Manipulação de Objetos através de um Modelo de Dados Temporal; XIII Congresso da SBC; Florianópolis; p.; 1993.
- [13] Snodgrass, R.; The Temporal Query Language TQUEL; ACM Trans. On Database Systems; v.12; n.2; june; pp.247-298; 1987.
- [14] Sarda, N. L.; Extensions to SQL for Historical Databases; IEEE Transaction on Knowledge and Data Engineering; v.2; n.2; june; pp. 220-230; 1990.
- [15] Sarda, N. L.; Algebra and Query Language for A Historical Data Model; The Computer Journal; v.33; n.1; pp. 11-18; 1990.
- [16] Sarda, N. L.; Extensions to SQL for Historical Databases; IEEE Transaction on Knowledge and Data Engineering; v.2; n.2; june; pp. 220-230; 1990.
- [17] Sarda, N. L.; Algebra and Query Language for A Historical Data Model; The Computer Journal; v.33; n.1; pp. 11-18; 1990.