

# TVM WeB - Uma Interface Visual para o Modelo Temporal de Versões

Vincent Nelson Kellers da Silveira, Nina Edelweiss e Renata de Matos Galante

Instituto de Informática - Universidade Federal do Rio Grande do Sul  
Caixa Postal 15.064 - 91.501-970 – Porto Alegre - RS - Brazil

{vincent, nina, galante}@inf.ufrgs.br

**Abstract.** *This paper describes TVM WeB: Temporal Versions Model Web Browser, a graphical browsing and querying tool, developed for the Temporal Versions Model (TVM). TVM WeB aims to supply the user with a friendly and easy learning interface that explores the temporal and versioned information stored in a TVM database, as well as TVM databases own properties.*

**Resumo.** *Este trabalho apresenta o TVM WeB: Temporal Versions Model Web Browser, uma ferramenta gráfica de navegação e consulta desenvolvida para o Modelo Temporal de Versões (TVM), que visa fornecer ao usuário uma interface amigável, de fácil aprendizado que explora as informações temporais e versionadas contidas em uma base TVM, bem como propriedades de bases de dados deste modelo.*

## 1. Introdução

As mais diversas aplicações, tais como das áreas científica, financeira, administrativa e comercial necessitam armazenar em bases de dados a evolução temporal dos seus dados. A evolução dos dados destas aplicações é implementada através do uso de modelagem temporal e de bases de dados que suportem esta modelagem.

Igualmente importante para diversas aplicações é o conceito de versão, que modela diferentes estágios da evolução de uma mesma entidade em tempos distintos ou sob diferentes pontos de vista. A técnica de versionamento possui aplicação em áreas como CAD (*Computer Aided Design*), CASE (*Computer Aided Software Engineering*) e SCM (*Software Configuration Management*), permitindo o trabalho cooperativo e a configuração de alternativas para projetos de software [Moro 2002], por exemplo.

A combinação da modelagem temporal e versionada permite acesso aos valores alterados ao longo do tempo nas diversas versões de uma entidade, inclusive as modificações descartadas, sendo que este acesso pode ser restrito a um determinado período no tempo. Tal resultado não pode ser alcançado com uso exclusivo da modelagem de versões, posto que as dimensões temporais, fundamentais para que esta flexibilidade de acesso seja atingida, não são tratadas pelo modelo de versões.

O TVM (*Temporal Versions Model*) [Moro 2002] incorpora ao conceito de versão o conceito de tempo, a fim de permitir o acompanhamento da evolução temporal dos objetos, versionados ou não, armazenados em uma base de dados.

Ferramentas gráficas unem as capacidades de grandes bases de dados e linguagens de consulta e disponibilizam ao usuário a possibilidade de construção de

buscas e navegação em bases de dados através de interfaces que abstraem o conhecimento técnico requerido por interfaces textuais. A fim de aumentar o rendimento de usuários com pouca familiaridade com a ferramenta, usualmente são oferecidos recursos que simulam graficamente a construção de consultas e a exibição de esquemas. Por outro lado, tais ferramentas gráficas oferecem a possibilidade de utilização de linguagens de consulta, permitindo a usuários já experimentados aumento de desempenho na execução de suas tarefas.

Este trabalho descreve o navegador visual do Modelo Temporal de Versões (TVM – *Temporal Versions Model*). Esse navegador foi desenvolvido com base nas características específicas do TVM, em estudos sobre as possíveis formas de representação de informações armazenadas em bancos de dados e na análise das funcionalidades de ferramentas visuais para navegação e consulta em bases de dados.

O objetivo deste trabalho foi produzir uma interface visual desenvolvida em ambiente web para navegação e consulta a uma base de dados TVM. A opção pelo ambiente web ocorreu devido à larga utilização de aplicações instaladas neste ambiente, que oferece independência de plataforma por parte do cliente e facilidade na alteração do formato de apresentação dos dados ao usuário. A aplicação possui formato visual, pois esta representação facilita a compreensão dos dados por parte do usuário.

Este trabalho encontra-se dividido em 5 seções. A seção 2 apresenta o Modelo Temporal de Versões. A seção 3 apresenta o TVM WeB e sua arquitetura. Na seção 4 é realizada uma comparação da ferramenta desenvolvida com outros trabalhos. Na seção 5 são apresentadas as conclusões do artigo e trabalhos futuros.

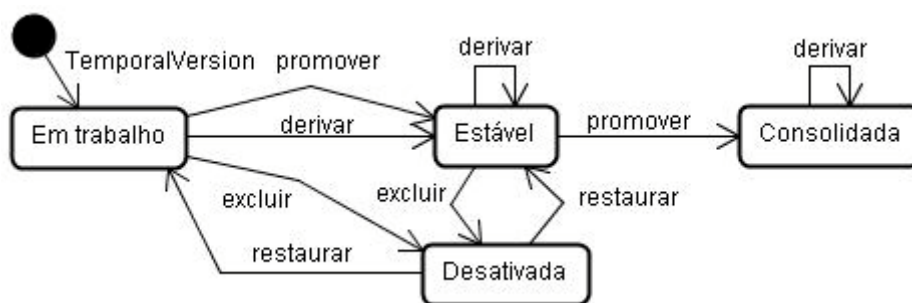
## **2. Modelo Temporal de Versões**

O Modelo Temporal de Versões - TVM (*Temporal Versions Model*) [Moro 2002] permite armazenar o histórico das modificações realizadas nas diferentes versões dos dados, além dos tempos de vida de cada versão.

Os elementos que podem ser temporalizados no TVM são objetos, versões, atributos e relacionamentos. Como um objeto possui uma linha de tempo para cada uma de suas versões, e um objeto pode possuir diversas versões, geram-se duas ordens de tempo: o tempo ramificado e o tempo linear. O tempo ramificado refere-se às diferentes linhas de tempo das versões de um objeto, e o tempo linear corresponde à evolução temporal de cada versão. A temporalidade é representada através de um rótulo de tempo que possui variação discreta e modela duas dimensões temporais: o tempo de validade e o tempo de transação [Jensen 1998].

A hierarquia do TVM permite ao usuário definir dois tipos de classes: classe de aplicação não temporal e não versionável, uma classe que não possui definição de dados versionados nem temporalizados e classe de aplicação temporal e versionada, uma classe onde atributos e relacionamentos podem ser definidos como estáticos ou temporalizados. Instâncias desta classe, que podem ser objetos não versionados, objetos versionados e versões, possuem um rótulo de tempo associado.

O TVM prevê regras utilizadas nas operações de atualização da base de dados, que visam garantir que o tempo de vida de uma versão estará contido no tempo de vida do objeto versionado, ao passo que o tempo de vida de um objeto, seja este objeto um atributo ou relacionamento, estará contido no tempo de vida da versão.



**Figura 1. Diagrama de estados de uma versão**

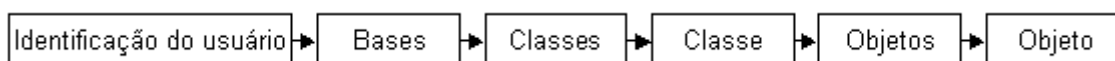
Toda versão possui um *status* associado, sendo que este *status* pode ser modificado ao longo da vida de uma versão devido a eventos especiais exibidos na Figura 1. Estes *status* são:

- em trabalho - estado que uma versão recebe ao ser criada na base de dados. Neste *status*, além de ser considerada temporária, uma versão pode ser apagada e alterada;
- estável - versões nesse *status* não podem ser mais alteradas, mas podem ser logicamente apagadas. Versões em trabalho atingem o *status* estável quando são promovidas pelo usuário ou quando usadas na derivação de uma nova versão;
- consolidada - neste *status* as versões não podem ser apagadas nem modificadas. Uma versão é consolidada através de uma ação explícita do usuário;
- desativada - a versão pode ser apenas consultada e restaurada. Representa uma versão excluída logicamente.

### 3. Arquitetura e funcionalidades do TVM WeB

O modelo de arquitetura utilizado na construção do TVM WeB é o MVC - *Model View Controller* [Gamma 1995]. Neste modelo, os *beans* são classes que representam objetos de uma realidade modelada. Os *controllers* são responsáveis pelo fluxo de navegação do usuário da aplicação, identificando que ações ele pode tomar, sendo os *modellers* responsáveis por implementar as ações disponibilizadas ao usuário.

A ferramenta desenvolvida neste trabalho utiliza a base de dados IBM DB2, escolhida por ser um banco de dados que opera de acordo com o modelo objeto relacional e por ser o mais próximo ao SQL-92 no suporte aos tipos de dados temporais. A ferramenta tem suas funcionalidades orientadas para a web, sendo o fluxo de ativação dos diferentes navegadores do TVM WeB apresentado na Figura 2.



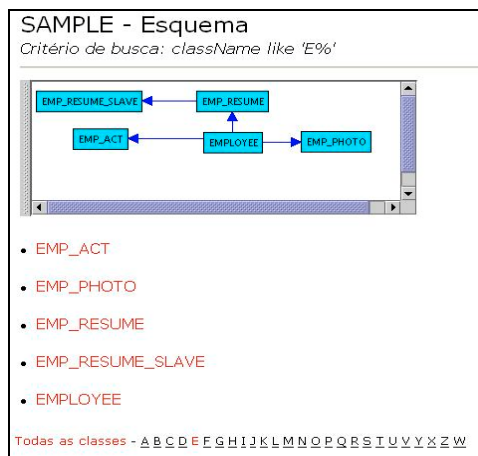
**Figura 2. Fluxo de utilização dos navegadores**

O TVM WeB possui um mecanismo de identificação do usuário que está iniciando a sessão de trabalho. Este dispositivo visa restringir apenas a determinadas bases de dados o acesso de usuários habilitados a utilizar a ferramenta.

Ao iniciar uma sessão de trabalho na ferramenta, o usuário deve selecionar no navegador de bases de dados a base que deseja visualizar, de acordo com as restrições

de acesso aplicadas a cada usuário cadastrado na base de dados. Após a seleção de uma determinada base de dados, o navegador de classes é invocado.

O navegador de classes, ilustrado na Figura 3, permite que o esquema de uma base de dados seja visualizado graficamente, em formato semelhante àquele utilizado em diagramas de classe. Este navegador ainda oferece ao usuário a possibilidade de selecionar uma determinada classe e visualizar os seus detalhes.



**Figura 3. O navegador de classes**

O navegador de classe, ilustrado na Figura 4, permite ao usuário a identificação dos atributos de uma classe, de classes relacionadas, das operações implementadas pela classe e permite a exibição dos objetos da classe selecionada, com destaque para atributos e relacionamento temporalizados. A exibição de objetos pode ser restringida através da especificação de um critério de busca SQL.

SAMPLE - Classe EMPLOYEE - Detalhes

Salvar configurações

Atributos [Esconder](#) | [Acima](#) | [Abaixo](#)

- EMPNO
- (t) SALARY
- STATUS

Relacionamentos [Esconder](#) | [Acima](#) | [Abaixo](#)

- (t) EMP\_RESUME

Objetos [Esconder](#) | [Acima](#) | [Abaixo](#)

[Exibir os objetos](#) - (32 objetos)

Especificar critério de busca

[Submeter busca](#)

Operações [Esconder](#) | [Acima](#) | [Abaixo](#)

- public void Metodo1 (String Parametro1 , Vector Parametro2 )

**Figura 4. Navegador de classe**

O navegador de objetos exibe todos os objetos de uma classe, ou ainda, objetos que atendem a um critério de busca, com destaque para os objetos versionados. Os atributos temporalizados possuem o seu valor corrente exibido tanto neste quanto no navegador de objeto. Por utilizar representação textual dos objetos, o navegador permite

ao usuário definir a ordem de exibição dos atributos e a ordenação em que os próprios objetos são exibidos. As configurações de interface realizadas por cada usuário são persistidas, para que em uma próxima sessão de trabalho sejam reutilizadas.

O navegador de objeto, ilustrado na Figura 5, permite ao usuário visualizar a evolução temporal dos atributos de um objeto, a sua posição dentro da hierarquia de versões, realizar navegação sincronizada nos objetos relacionados ao objeto visualizado, além de permitir seleção de outros objetos do grafo de versões, a fim de que estes sejam analisados em igual nível de detalhe.

SAMPLE - Classe EMPLOYEE - Detalhes do objeto

Salvar configuração

Atributos Esconder | Acima | Abaixo

Empno: 000010  
 (€) Salary: 52750,00  
 Status:

Dados temporais Esconder | Acima | Abaixo

SALARY (2 registro(s)) Esconder

Valor	TVI	TVF	TTI	TTF
100	01/01/2003 12:00:00	01/02/2003 02:00:00	01/01/2003 12:00:00	01/01/2003 12:00:00
1000	01/02/2003 12:00:00		15/01/2003 12:00:00	15/01/2003 12:00:00

Versões Esconder | Acima | Abaixo

\*O item em vermelho indica a versão corrente do objeto versionado.

Objetos relacionados Esconder | Acima | Abaixo

EMP\_ACT: 3 Exibir seção

**Figura 5. O navegador de objeto**

No navegador de classe e de objeto é permitido ao usuário definir quais seções serão visíveis bem como em que ordem as diferentes seções do navegador serão exibidas. As configurações definidas por cada usuário da ferramenta são persistidas a fim de serem reutilizadas nas sessões de trabalho seguintes.

#### 4. Comparação com outros trabalhos

Apresentado em [Agrawal 1990], o OdeView é um navegador que permite a visualização gráfica de uma base de dados Ode, com suporte a navegação sincronizada e a consultas definidas através do preenchimento de formulários ou através de um modo semelhante àquele implementado pelo QBE.

PESTO [Carey 1996] é um navegador independente de bases de dados, com funcionalidades semelhantes às do OdeView. O PESTO diferencia-se nas consultas, pois estas utilizam um paradigma de buscas integradas à visualização de dados. PESTO permite que critérios de busca como seleção, negação e disjunção sejam utilizados sem que o usuário escreva código explícito de consulta.

A ferramenta definida em [Silva 1998] opera com bancos de dados versionados, e utiliza o conceito de navegadores com interfaces híbridas especializados em diferentes porções de uma base de dados.

O TVM WeB utiliza conceitos introduzidos por estas ferramentas, como a navegação sincronizada, a especificação de critérios de busca integrada à consulta de dados e navegadores especializados em diferentes porções de uma base de dados.

Comparado a estas ferramentas, o TVM WeB diferencia-se por ser voltado para bancos de dados TVM explorando as características de temporalidade e versionamento destas bases e por utilizar o ambiente web, que além de possuir aplicações muito conhecidas atualmente, garante independência de plataforma por parte do usuário.

## **5. Conclusões e trabalhos futuros**

Composto por diferentes navegadores especializados e com funcionalidades extraídas de ferramentas semelhantes, o TVM WeB permite ao usuário não familiarizado com navegadores de bases de dados o acesso às informações armazenadas em bases TVM, como a visualização da evolução temporal e das versões de objetos, sem que consultas específicas para tanto sejam escritas, bem como a visualização de objetos não versionados e não temporalizados. Visando facilidade de uso, a ferramenta é desenvolvida em ambiente web, uma vez que aplicações baseadas neste ambiente estão amplamente difundidas. Devido à limitação de espaço, apenas as principais características da ferramenta estão detalhadas neste trabalho, onde é apresentado um resumo do estudo que compara o TVM WeB com outras ferramentas existentes.

Após a implementação inicial do TVM WeB, algumas questões persistem, como a utilização de interfaces gráficas para consultas, o suporte à linguagem TVQL (linguagem de consulta do TVM, definida em Moro, 2002) nos navegadores de objeto e de objetos e a persistência do posicionamento do grafo de classes.

## **Referências**

- AGRAWAL, R.; GEHANI, N. H.; SRINIVASAN, J. OdeView: The Graphical Interface to Ode. SIGMOD Record, New York, v.9, n.2, June 1990. ACM SIGMOD International Conference on Management of Data, 1990.
- CAREY, M. et al. PESTO: An Integrated Query/Browser for Object Databases. In: VLDB CONFERENCE, 22., 1996, Mubai, India. Proceedings.
- GAMMA, E. et al. Design Patterns. Addison-Wesley Pub Co., 1995.
- JENSEN, C.S. et al. The Consensus Glossary of Temporal Database Concepts February 1998 Version. In: ETZION, O. et al. Temporal Databases Research and Practice. Berlin-Heidelberg: Springer-Verlag, 1998. p. 367-405.
- MORO, M. M.; EDELWEISS, N.; SANTOS, C. S. Modelo Temporal de Versões. IX Concurso de Tesis de Maestría Clei-UNESCO, Primeiro lugar, XXVIII Conferencia Latinoamericana de Informatica, November 2002, Montevideo, Uruguay, pp.116.
- SILVA, J. T. Uma Interface Visual para Modelos de Bancos de Dados Orientados a Objeto com Suporte para Versões. Porto Alegre: PPGC da UFRGS, 1998. Dissertação de Mestrado.