

Implementação de um Mecanismo para Gerência de Evolução de Esquemas de DW

Mauro Sérgio da Costa Leal, Nina Edelweiss

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{mcosta, nina}@inf.ufrgs.br

Resumo. *Este trabalho descreve o projeto e a implementação de uma ferramenta utilizada para especificar esquemas de um data warehouse (DW) com suporte ao gerenciamento de versões, permitindo o controle e o armazenamento do histórico da evolução dos esquemas e dos dados (instâncias). A ferramenta utiliza como base um modelo conceitual genérico de evolução de DW. A ferramenta permite especificar esquemas de DW e executar operações de modificação e consulta sobre o esquema e sobre as versões do esquema. Além disso, a cada nova versão do esquema a ferramenta automaticamente executa a adaptação dos dados existentes no DW. Cabe ressaltar ainda que, qualquer sistema externo pode fazer a carga dos dados no DW, sendo geradas automaticamente informações temporais que controlam o período de validade e as versões das instâncias do DW.*

1 Introdução

Durante o processo evolutivo de um DW as alterações no esquema ocorrem com uma frequência muito maior do que as alterações nas instâncias. Alterações no esquema costumam ser complexas e geram um grande impacto no DW. Pequenas mudanças no esquema implicam em muitas operações de criação, alteração e exclusão de um grande volume das instâncias. Essas operações precisam ser executadas para adaptar os dados armazenados no esquema antigo ao novo esquema.

Ao longo do tempo foram propostos diversos modelos de DW com suporte à evolução de esquemas (Blaschka, 1999; Vaisman, 2001; Vaisman e Mendelzon, 2002) e à evolução dos dados (Hurtado, Mendelzon e Vaisman (1999, 2000, 2002); Eder e Koncilia (2001, 2002 e 2003); Koncilia, 2003; Koncilia et. al, 2004). Devido à importância da manutenção dos históricos e da correta adaptação das instâncias na alteração das dimensões, o número de trabalhos na área de evolução dos dados é maior que o na de evolução das instâncias. Poucos trabalhos oferecem suporte tanto à evolução do esquema quanto à evolução das instâncias.

O Modelo de Gerenciamento de Versões para Evolução de DW proposto por Alexandre Schlöttgen (SCHLÖTTGEN, 2004) oferece suporte tanto à evolução de esquemas quanto à evolução das instâncias. Este modelo apresenta definições para cada componente de um esquema de DW (cubos, dimensões, métricas, etc.) com suporte a múltiplas versões. Apresenta, ainda, um conjunto de operações sobre o esquema e sobre as instâncias, conjunto este que cobre todos os tipos de modificações possíveis em um DW. Além disso, são definidas operações sobre as versões do esquema e mecanismos para manter o controle de validade das instâncias do DW.

O objetivo deste artigo é apresentar o projeto, a especificação e a implementação de uma ferramenta que proporcione as principais funcionalidades de um DW com gerenciamento de versões, especificadas no modelo conceitual de Schlöttgen (SCHLÖTTGEN, 2004). As principais funcionalidades da ferramenta são: a) permitir a especificação de esquemas de DW; b) armazenar e controlar as versões do esquema; c) possibilitar consultas às versões anteriores do esquema; d) controlar automaticamente o período de validade e a versão dos dados inseridos no DW.

O artigo está organizado da seguinte forma. A seção 2 apresenta os principais trabalhos relacionados na área de evolução de DW. A seção 3 descreve resumidamente o modelo conceitual utilizado como base para a implementação da ferramenta. A seção 4 trata da implementação e dos módulos da ferramenta. Finalmente, na seção 5 são apresentadas as conclusões e perspectivas futuras.

2 Modelo para Evolução de Data Warehouse

Esta seção apresenta o Modelo de Gerenciamento de Versões para Evolução de *Data Warehouses*, proposto por Schlöttgen (SCHLÖTTGEN, 2004). Este modelo foi escolhido como base para a implementação da ferramenta porque possui suporte tanto à evolução de esquemas quanto à evolução dos dados. Além disso, apresenta soluções para manter bom desempenho e alto nível de disponibilidade, que são requisitos indispensáveis para um DW ter sucesso.

O modelo apresenta uma entidade especial denominada “Versão do Esquema”, que possui um identificador único utilizado para referenciar as diferentes versões dos esquemas e dois atributos temporais (data inicial e data final) para delimitar o intervalo de vigência da referida versão, ou seja, o intervalo de tempo durante o qual a versão pode receber atualizações.

Uma versão pode assumir os seguintes estados: em trabalho, atual, ativa ou congelada. As principais operações que podem ser executadas sobre uma versão são: derivação, ativação, congelamento e exclusão física.

As versões seguem obrigatoriamente uma seqüência de criação linear, possibilitando somente a derivação de novas versões a partir da última versão criada, que é chamada de versão atual. A única exceção acontece quando um novo esquema é criado, pois neste caso não existe uma versão antecessora para ser derivada. Uma nova versão sempre inicia no estado “em trabalho”.

A seguir, é apresentada uma breve descrição das operações que podem ser executadas sobre as versões dos esquemas e as mudanças de estado que ocorrem em cada uma destas operações. A Figura 1 apresenta o diagrama de estados de uma versão que ilustra a transição de estados durante a execução das operações:

- operação de derivação da versão – cria uma nova versão idêntica ao esquema atual, mas sem dados populados. Esta nova versão inicia no estado “em trabalho”;
- operação de ativação da versão – pode ser aplicada apenas sobre uma versão “em trabalho”. A versão “atual” passa a ser “ativa” e a versão “em trabalho” passa a ser “atual”. Esta operação gera uma lista de criação e alteração de esquema que é utilizada para traduzir o modelo especificado na versão para a linguagem do SGBD que armazena o DW. Também são gerados scripts que buscam dados da versão imediatamente anterior, fazem as adaptações necessárias e populam os dados da versão que passou para o estado “atual”;
- operação de congelamento da versão – consiste na atribuição da data corrente para a data de fim de vigência da versão. As versões congeladas não são mais vistas e acessadas pela maioria dos usuários do DW;
- operação de exclusão física da versão em trabalho – remove a versão do esquema sem guardar o histórico das suas definições. Executada sobre versões “em trabalho” que não serão mais utilizadas.

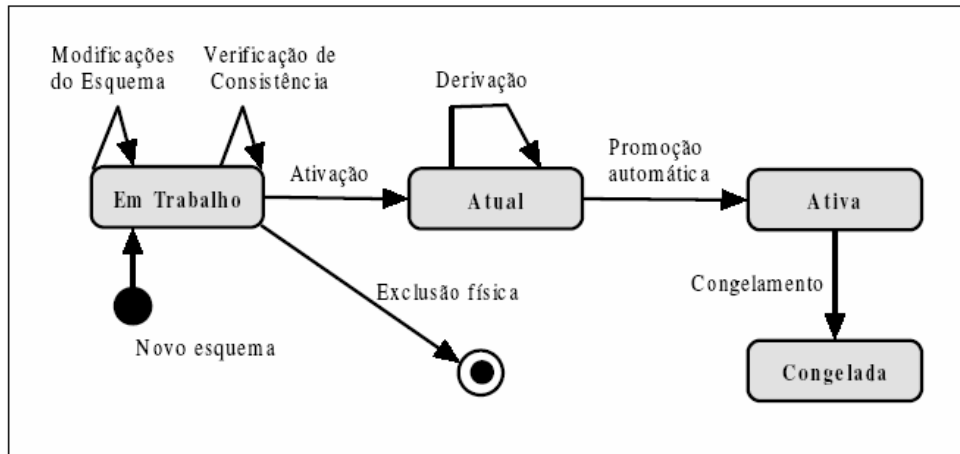


Figura 1: Diagrama de estados de uma versão (SCHLÖTTGEN, 2004).

3 Descrição da Ferramenta

Esta seção descreve a ferramenta Gerenciador de Versões de Data Warehouse – GVDW (LEAL, 2005), que implementa as principais funcionalidades do Modelo de Gerenciamento de Versões para evolução de *Data Warehouses* proposto por Schlöttgen (SCHLÖTTGEN, 2004) e descrito na seção 2.

3.1 Implementação

A ferramenta foi implementada utilizando a ferramenta de desenvolvimento *Microsoft Visual Basic .NET 2003*. O sistema gerenciador de banco de dados (SGBD) e de DW utilizado foi o *Oracle Database 10g*. A ferramenta foi projetada e implementada utilizando o paradigma de orientação a objetos, o que possibilita a utilização de vários recursos que facilitam e organizam o desenvolvimento, como por exemplo, herança, polimorfismo e reutilização de código. As versões do esquema que são especificadas pelo usuário ficam armazenadas em um repositório de esquemas. Estas versões podem ser consultadas e alteradas. Sobre elas também podem ser executadas algumas operações como derivação, ativação, congelamento que provocam alterações no esquema e nos dados relacionados a estes esquemas.

A Figura 2 mostra uma visão geral da arquitetura do sistema. Inicialmente o usuário define um esquema inicial (1). A partir deste, ele pode definir outros esquemas (1) e visualizar/alterar esquemas anteriores (2). Quando o usuário executa uma operação sobre uma versão de esquema (3), o “Gerenciador de Versões do Esquema” gera as atualizações necessárias no esquema (4) que são executadas pelo “Gerenciador de BD” no repositório de esquemas (5). Quando necessário, o “Gerenciador de Versões do Esquema” também comunica ao “Gerenciador de Versões de Dados” que são necessárias manipulações de dados (6). O “Gerenciador de Versões de Dados” gera as alterações necessárias (7) e passa para o “Gerenciador de BD” executá-los no repositório de dados (8).

Quando outros sistemas alteram o repositório de dados, os gatilhos gerados pela ferramenta atualizam os atributos de controle de versão e a data inicial de validade das instâncias (9). Quando dados são inseridos, a data inicial de validade da instância e a versão para a qual este dado pertencente são preenchidos automaticamente.

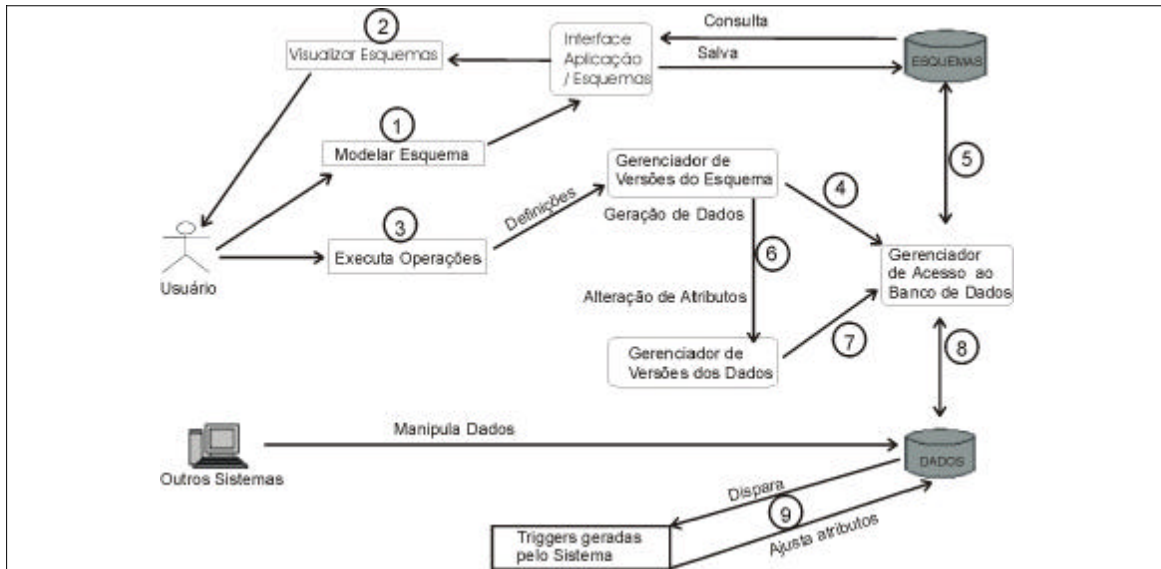


Figura 2: Arquitetura e funcionamento da ferramenta.

3.2 Módulos da Ferramenta

A ferramenta é composta pelos seguintes módulos:

- **Interface Aplicação / Esquemas** – este módulo é responsável pela comunicação entre a aplicação e o repositório onde são armazenadas as versões do esquema. Contém métodos para salvar uma versão no repositório de versões do esquema, bem como para consultar os dados das versões do esquema já armazenadas, quando a aplicação necessita mostrar a versão de um esquema previamente criado;

- **Gerenciador de Versões do Esquema** – este módulo é responsável por controlar o versionamento propriamente dito. Controla as operações que são executadas sobre as versões – ativação, derivação, etc. – realizando as consistências e ações necessárias. Utiliza os métodos dos módulos “Gerenciador de Acesso ao Banco de Dados” e “Gerenciador de Versões dos Dados” para executar estas operações;

- **Gerenciador de Versões dos Dados** – este módulo é o responsável por gerenciar a população dos dados de uma nova versão, quando esta é ativada. Os dados são buscados do repositório de dados da versão imediatamente anterior, adaptados, caso seja necessário, e inseridos no repositório de dados da versão atual. A Figura 3 mostra um exemplo de um script de adaptação e população de dados entre duas versões;

```

insert into ESQ_VENDAS_2
(CIDADE_ID, MODELO_ID, QUANTIDADEVENDIDA, VALORVENDIDO )
select ESQ_VENDAS_1.CIDADE_ID, ESQ_VENDAS_1.MODELO_ID, sum(QUANTIDADEVENDIDA), sum(VALORVENDIDO)
from ESQ_VENDAS_1 inner join ESQ_VENDAS_1_GV on ESQ_VENDAS_1.CIDADE_ID = ESQ_VENDAS_1_GV.CIDADE_ID and ESQ_VENDAS_1.MODELO_ID = ESQ_VENDAS_1_GV.MODELO_ID and ESQ_VENDAS_1.DATA_ID = ESQ_VENDAS_1_GV.DATA_ID
group by ESQ_VENDAS_1.CIDADE_ID, ESQ_VENDAS_1.MODELO_ID

```

Figura 3: Exemplo de script de adaptação e população dos dados.

- **Gerenciador de Acesso ao Banco de Dados** – este módulo é o responsável por traduzir o esquema lógico especificado em uma versão para a linguagem do SGBD onde os dados do DW são efetivamente armazenados. A Figura 4 mostra o exemplo de um script que traduz uma dimensão especificada em uma versão do esquema para a linguagem do SGBD. O módulo também possui duas outras funções importantes. Uma delas é gerar o script de adaptação dos dados que são solicitados pelo módulo “Gerenciador de Versões dos Dados”. A outra função é gerar os gatilhos nas tabelas de fato e de dimensão que fazem com que os atributos de gerenciamento de versão dos dados sejam preenchidos automaticamente quando é executada uma operação de inserção (*insert*) no repositório de dados.

```

CREATE DIMENSION "DM_TEMP01_1"
LEVEL "DATA" IS "ESQ_TEMP01_1"."DATA_ID"
LEVEL "SEMANA1" IS "ESQ_TEMP01_1"."SEMANA1_ID"
LEVEL "MES" IS "ESQ_TEMP01_1"."MES_ID"
HIERARCHY "COMPLETA"
("DATA" CHILD OF
"SEMANA1" CHILD OF
"MES")
ATTRIBUTE "DATA" DETERMINES (DIA )
ATTRIBUTE "SEMANA1" DETERMINES (SEM1NUMERO )
ATTRIBUTE "MES" DETERMINES (MESEDESCR )

```

Figura 4: Exemplo de script de criação de dimensão.

4 Trabalhos relacionados

Vários trabalhos foram propostos na área de evolução de dados de DW. Hurtado, Mendelzon e Vaisman (1999, 2000, 2002) apresentam o *Dimensional Update Model*, um modelo conceitual para o controle das alterações das dimensões que assegura a consistência semântica dos dados e das estruturas multidimensionais durante as alterações. Eder e Koncilia (2001, 2002 e 2003) definem o *Modelo Temporal de DW*, que consiste de um modelo matemático para o controle das operações de alteração da constituição das dimensões via operações matriciais. Mais tarde foi proposto o *Modelo Bi-temporal de Data Warehouse* (Koncilia, 2003), que é uma extensão ao modelo anterior e armazena além do tempo de validade dos dados, também o tempo de transação. Posteriormente, foi proposto o *Modelo de um Data Warehouse Multiversionado* (Koncilia et. al, 2004), que oferece suporte à evolução de dados e esquemas.

Também existem trabalhos importantes na área de evolução de esquemas. Em (Blaschka, 1999) é proposto um modelo conceitual que descreve evoluções de um esquema multidimensional e seus efeitos sobre os esquemas e as instâncias, apresentando uma álgebra de evolução de esquemas e como esta álgebra deve ser usada para habilitar uma ferramenta a utilizar evolução de esquemas. Em (Vaisman, 2001; Vaisman; Mendelzon, 2002) é proposto um modelo multidimensional temporal para o gerenciamento das várias versões dos dados e esquemas das dimensões, focando principalmente as relações hierárquicas entre os membros.

Em contraste com os trabalhos apresentados acima, o modelo proposto por Schlöttgen e projetado e implementado neste trabalho é mais abrangente visto que trata da evolução de esquemas e de dados no mesmo modelo. Além disso, este modelo de versões pode ser utilizado para controlar a evolução de outros modelos multidimensionais. É o caso deste trabalho, onde o modelo de gerenciamento de versões é utilizado para controlar a evolução de um modelo estrela, que é o modelo lógico utilizado na ferramenta de DW da *Oracle*. O modelo estrela é o modelo de relacionamento de um fato com suas dimensões, onde na sua representação gráfica a tabela de fato fica no centro e as tabelas de dimensões nas pontas, lembrando o formato de uma estrela. Este modelo possui as tabelas de dimensões desnormalizadas. Foi o modelo escolhido na implementação do trabalho por ser simples e por apresentar um ganho de performance devido ao pequeno número de junções realizadas nas consultas.

5 Conclusões

Este artigo descreveu o projeto e a implementação de uma ferramenta que oferece suporte ao armazenamento de versões do esquema e dos dados de um DW. A ferramenta implementa as definições de um DW versionado propostas por Schlöttgen em (SCHLÖTTGEN, 2004) que podem ser aplicadas a um modelo estrela.

Foram implementadas as operações de evolução de esquema que permitem inserir ou excluir um componente no DW. As operações de alteração de esquema não foram implementadas, mas podem ser obtidas utilizando a combinação das operações de

exclusão e inserção. As operações de evolução das instâncias não foram implementadas, visto que o SGBD disponibiliza tais operações.

Quanto a evolução dos dados, em operações de inserção, que são as mais utilizadas em DW, a data inicial de validade da instância e a versão para o qual o dado pertence são automaticamente gerenciados por mecanismos criados pela ferramenta. Como limitação, a ferramenta ainda não cria automaticamente mecanismos para controlar a versão e o período de validade das instâncias nas operações de atualização (*update*) e exclusão (*delete*). Estes mecanismos podem ser criados gerando gatilhos para os eventos de *update* e *delete* para cada um das tabelas de fato e de dimensão de cada versão que for ativada.

Com relação à extração dos dados, não foi desenvolvida nenhuma ferramenta de consulta. As consultas devem ser feitas por outras ferramentas de extração de dados. Entretanto, essas ferramentas não oferecem suporte a múltiplas versões, cabendo ao usuário controlar a versão dos dados que deve ser consultada. Como trabalhos futuros, uma solução é desenvolver uma camada conversora de consultas que atue entre o DW e as aplicações que o acessam. A principal tarefa desta camada é resolver as referências às versões de esquemas e dados, localizando e acessando os dados da versão mais adequada segundo os critérios das consultas. Dentro desse contexto, uma outra abordagem seria utilizar visões materializadas em conjunto com o modelo de gerência de versões.

6 Referências Bibliográficas

- BLASCHKA, M.; SAPIA, C.; HÖFLING, G. On Schema Evolution in Multidimensional Databases. In: 1st INTERNATIONAL CONFERENCE ON DATA WAREHOUSING AND KNOWLEDGE DISCOVERY (DAWAK), Florence, Italy, 1999. **Proceedings...** Springer - LNCS v. 2676. p.153-164.
- HURTADO, C.; MENDELZON, A.; VAISMAN, A. Updating OLAP Dimensions. In: 3rd INTERNATIONAL WORKSHOP ON DATA WAREHOUSING AND OLAP, Kansas City, USA, 1999. **Proceedings...** ACM Press, p.60-66.
- HURTADO C.; MENDELZON A.; VAISMAN A. Maintaining Data Cubes Under Dimension Updates. In: 15th IEEE INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE), Sydney, Australia, 1999. **Proceedings...** IEEE Computer Society, p. 346-355.
- HURTADO C.; MENDELZON A. O. OLAP Dimension Constraints. In: SYMPOSIUM ON PRINCIPLES OF DATABASE SYSTEMS, Madison, USA, 2002. **Proceedings...** ACM, p. 169-179.
- KONCILIA C; EDER J. Changes of Dimension Data in Temporal DWs In: 3th INTERNATIONAL CONFERENCE ON DATA WAREHOUSING AND KNOWLEDGE DISCOVERY (DAWAK), Munich, Germany, 2001. **Proceedings...** Springer - LNCS 2113, 284-293.
- KONCILIA C.; EDER J.; MORZY, T. The COMET Metamodel for Temporal DWs. In: 14th INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION SYSTEMS ENGINEERING (CAiSE), Toronto, Canada, 2002. **Proceedings...** Springer - LNCS 2348, p. 83-99.
- KONCILIA C.; EDER J.; MITSCHKE, D. Automatic detection of structural changes in DWs. In: 5th INTERNATIONAL CONFERENCE ON DATA WAREHOUSING AND KNOWLEDGE DISCOVERY (DAWAK), Prague, Czech Republic, 2003. **Proceedings...** Springer - LNCS 2737, p.119-128.
- KONCILIA, C. A Bi-Temporal DW Model. CAiSE, 2003.
- KONCILIA. C.; EDER, J.; BEBEL, B.; MORZY, T.; WREMBEL. R. Creation and management of versions in multiversion DW. In: SYMPOSIUM ON APPLIED COMPUTING, Nicosia, Cyprus, 2004. p. 717 – 723.
- SCHLÖTTGEN, A. **Modelo de Gerenciamento de Versões para Evolução de DWs**. 2004. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- VAISMAN, A. Updates View Maintenance and Time Management on Multidimensional Databases. Ph.D. Thesis - Universidad de Buenos Aires, 2001.
- VAISMAN, A.; MENDELZON, A. A Temporal Query Language for OLAP: Implementation and a Case Study. DATABASE PROGRAMMING LANGUAGES, 8TH INTERNATIONAL WORKSHOP (DBPL), Frascati, Italy, 2001., **Proceeding...** Springer – LNCS 2397, 2002, pages 78-96.
- LEAL, M. **Implementação de um Mecanismo para Gerência de Evolução de Esquemas de Data Warehouse**, 2005. Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.