

Título: Uma Proposta para Tratamento de Restrições de Integridade Semânticas em Sistemas de Informação.

Autores:

ALVARENGA, Geoflávia Guilarducci. geoflavia@inf.ufg.br
OLIVEIRA, Juliano Lopes. juliano@inf.ufg.br

Unidade Acadêmica: Instituto de Informática (INF-UFG).

Palavras chave: Regras de Negócio, Requisitos Funcionais, Restrições de Integridade.

1. Introdução

A construção de um Sistema de Informação (SI) envolve a especificação de dados e funções (ou processos) que manipulam esses dados. Tradicionalmente, esta especificação de dados e funções é feita, respectivamente, com base em técnicas da área de Bancos de Dados e de Engenharia de Software.

Na prática estas técnicas separadas não têm conseguido atender a demanda crescente de evolução dos SI. Esta demanda ocorre por diversos motivos, como por exemplo, mudanças nas legislações, alterações e costumes e padrões, novos pensamentos, novas tecnologias e tendências, entre outros.

Além disso, a utilização de técnicas isoladas não permite um tratamento uniforme para as Regras de Negócio (RN). Essas regras são fundamentais para o sucesso de SI, envolvendo declarações que definem ou restringem algum aspecto do negócio subjacente (GROUP, 2000).

A comunidade que pesquisa os SI defende a idéia de modelar, como uma parte independente da arquitetura do software, as regras de negócio. Esta proposta consiste em uma divisão do SI em três componentes: dados, funções e regras de negócio (GROUP, 2000; ZIMBRÃO, 2001; ZIMBRÃO, 2003).

Dessa forma, permite-se modelar de forma precisa uma determinada realidade. Por exemplo, é possível representar as RN conceitualmente, de maneira a facilitar o processo de compreensão das regras e de automatização da sua execução pelo SI.

O objetivo deste trabalho é apresentar a aplicação desta abordagem de modelagem de Regras de Negócio na construção de um SI complexo. Este SI atua no domínio de pecuária de corte, e está em fase final de desenvolvimento, no contexto de um projeto de pesquisa apoiado pelo CNPq (processos nº 505198/2004-5 e 507076/2004-4).

2. Metodologia

Existem algumas abordagens bastante difundidas para tratamento de Regras de Negócio. A comunidade de Bancos de Dados (BD) define RN através da criação de restrições de integridade em esquemas de BD (utilizando uma DDL - Linguagem de Definição de Dados). Esta abordagem deixa que o SGBD (Sistema Gerenciador de Bancos de Dados) realize todo o gerenciamento das RN.

A abordagem da Engenharia de Software define RN através da implementação de código específico (utilizando uma linguagem de programação). Assim, as RN são definidas e gerenciadas conjuntamente com as funções do sistema.

A proposta apresentada neste trabalho é uma terceira via, pois não utiliza fixa as RN no esquema do BD e tampouco as fixa nos programas do SI. A idéia consiste em chamadas dinâmicas de RN definidas fora do código do SI, utilizando a linguagem de regras OCL (*Object Constraint Language*) (WARMER, 2003; DEMUTH,1999; DEMUTH, 2001).

Neste contexto, a linguagem OCL tem um papel importante, porque ela permite a elaboração de modelos mais precisos e consistentes com o mundo real, isto é, o que se lê através da regra é o que se deseja expressar. OCL é *side-effect free*, ou seja, não muda o estado dos objetos do sistema, e também, permite realizar a transformação de modelos (WARMER, 2003; CORREA, 2004).

Assim, as funções do SI são completamente independentes das RN, podendo ser compiladas de forma isolada. Esta independência é a base das características gerais da solução aqui proposta, que reúne as seguintes vantagens:

- Estende o modelo conceitual de entidades do negócio do SI, tornando-o capaz de implementar regras de negócio que representam restrições de integridade semânticas.
- Mantém uma base de conhecimento documentada das regras de negócio do sistema. A formalização das regras é feita através de uma linguagem declarativa que torna possível gerar código de forma automática (ZIMBRÃO, 2001; ZIMBRÃO, 2003; DEMUTH,1999; DEMUTH, 2001).
- Permite que a aplicação controle a execução da verificação das regras de negócio através de chamadas a um componente de regras, e não através da incorporação das regras ao código do SI.
- Garante a separação das regras de negócio específicas das entidades de negócio, ou seja, não há regras espalhadas de maneira indisciplinada pelo código fonte (GROUP, 2000; ZIMBRÃO, 2001; ZIMBRÃO, 2003).

O mecanismo de tratamento das regras de negócio é dividido em duas fases:

1ª Fase - Implantação da regra: nesta fase é feito o armazenamento e a tradução da regra escrita em OCL para uma linguagem processável por computador.

A atividade de tradução consiste nos seguintes passos:

- a) obtém-se a regra de negócio armazenada como metadado associado a uma entidade de negócio;
- b) realiza-se a análise léxica e gramatical da regra através de um gerador de parser (GAGNON, 1998). Este gerador converte a regra numa estrutura de árvore, conhecida como AST (*Abstract Syntax Tree*) (HUSSMANN, 2000; FINGER, 2000);
- c) através da navegação na AST gerada no passo anterior, identifica-se qual a linguagem alvo para mapeamento do código OCL. Por exemplo, pode-se mapear as regras para SQL-DML (Linguagem de Manipulação de Dados) (MELTON, 2003);
- d) depois de identificado qual a linguagem alvo que será usada, constrói-se o código equivalente à regra em OCL. No caso da linguagem alvo ser SQL-DML, o resultado seria uma função armazenada no SGBD.

2ª Fase - Execução da regra: repetem-se os passos (a) e (b) da 1ª. Fase, e depois os passos a seguir:

- a) navega-se na AST e efetuam-se as substituições dos *tokens* (identificados como metadados de uma entidade de negócio) pelos valores das variáveis provenientes da aplicação (parâmetros das regras);
- b) a aplicação invoca a função que realiza o teste de verificação da regra

que rege a entidade de negócio.

Uma entidade de negócio pode possuir várias RN associadas. É recomendável efetivar a verificação das regras restritivas antes de alguma operação de manipulação sobre os dados da entidade – inserção, alteração ou exclusão, por exemplo.

3. Resultados

A separação das RN nos SI oferece resultados positivos. Em primeiro lugar, as regras não ficam espalhadas pelo código da aplicação. Com esta organização das regras, é possível conhecer e gerenciar as regras contidas no sistema, identificando as que se aplicam em cada conceito (entidade) de negócio. Além disso, há mais clareza para manutenção dessas regras.

Alguns trabalhos semelhantes à abordagem dada na solução deste trabalho, confirmam os resultados positivos que podem ser obtidos na construção de uma parte do software dedicado às regras de negócio, vejamos:

- Em (ZIMBRÃO, 2001; ZIMBRÃO, 2003) apresenta o projeto e a implementação da ferramenta Atena, que é uma máquina de execução de restrições escritas na linguagem OCL, construída sobre o topo de um SGBD Relacional. As regras em OCL são convertidas para a linguagem SQL, no nível de DML ou DDL. Esta ferramenta permite acoplar uma camada de regras de negócio numa aplicação já existente.
- Em (DEMUTH, 2001) apresenta uma investigação sobre como as regras escritas em OCL podem ser manipuladas em aplicações orientadas a objetos que fazem uso de SGBD's. Este trabalho explica o projeto da ferramenta OCL2SQL que consiste em um gerador de código SQL que faz a interpretação de um modelo de objetos na linguagem UML, obtém as regras escritas em OCL deste modelo, e gera um script em SQL, incluindo um esquema de BD em DDL, mais definições de visões e gatilhos.
- Em (DEMUTH, 1999) discute o uso da linguagem OCL para modelar restrições de integridade em BD, usando a linguagem SQL, e propõe um conjunto de padrões que descrevem o mapeamento de expressões em OCL para código em SQL.

Os trabalhos citados (ZIMBRÃO, 2001; ZIMBRÃO, 2003; DEMUTH, 1999; DEMUTH, 2001) ratificam o fato de que é possível a construção de um mecanismo para gerenciamento de regras de negócio que envolve a representação das regras em linguagem formal (como OCL), bem como, gerar o código fonte para a implantação da regra, e permitir a verificação da consistência dessas regras sob o controle da aplicação ou de um SGBD específico.

Nosso trabalho segue a mesma sistemática dos trabalhos comentados, porém com algumas variações, como por exemplo:

- Primeiramente, não possui o porte completo de uma máquina de regras como é feito nos trabalhos de (ZIMBRÃO, 2001; DEMUTH, 2001).
- O mapeamento da regra em OCL para SQL usa, principalmente, a criação de funções armazenadas em um SGBD.
- O acionamento das funções armazenadas é controlado pela aplicação, ou seja, a aplicação é que decide quando deverá realizar o teste de verificação da regra para validar a consistência dos dados. Isso se deve ao fato que, cada aplicação pode ter sua própria necessidade de quando

suas regras de negócio deverão ser verificadas, pois podem valorizar mais o requisito de desempenho ao de restrição total. Diante disso, optamos por não usar a verificação de restrições imediatas.

- Em (DEMUTH, 2001) propõe, basicamente, a aplicação da verificação da regra de negócio através da avaliação de uma visão de BD que retorna tuplas inválidas. Neste caso, entendemos que quando uma tupla é atualizada no BD (isto é, quando ocorre alguma transação de inserção, alteração ou remoção de tupla), imediatamente antes dessa transação ser efetivada, é executada a verificação das regras através de uma visão que retorna todas tuplas que violam essas regras. Em nosso mecanismo não é necessário que uma tupla seja atualizada, pois ele verifica se a futura tupla viola as regras incidentes sobre a referida tupla, e informa para aplicação este resultado, quando solicitado.

Como as regras estão escritas em uma linguagem que não permite mudança de estado dos objetos, a verificação da regra não causa nenhum efeito colateral no sistema; ao contrário, a regra poderá auxiliar na descoberta de possíveis defeitos ou inconsistências com o mundo real.

O mecanismo proposto foi projetado para atender as características de um SI específico. Porém, a abordagem pode ser utilizada em outros SI, já que a base da proposta é a definição de regras de negócio para as entidades de negócio independentemente do código da aplicação específica. Esses conceitos básicos certamente poderão ser portados para outros SI.

3. Conclusões

Há muitas vantagens na implementação de regras de negócio independente das aplicações ou dos bancos de dados utilizados pelos SI. Neste sentido, a proposta deste trabalho concentra suas contribuições em duas áreas da Computação.

Sob a perspectiva da Engenharia de Software, regras de negócio são denominadas de requisitos funcionais e modeladas com a UML. Neste trabalho, a modelagem conceitual das regras é feita através da UML (*Unified Modeling Language*), mas combinada com uma extensão (chamada OCL). Esta combinação oferece benefícios significativos na compreensão dos modelos de regras do SI.

Sob a perspectiva de Bancos de Dados, as regras de negócio são conhecidas como restrições de integridade. Este trabalho permite tratar um importante subconjunto dessas restrições, que são as restrições semânticas. A contribuição do trabalho é, através da conversão de regras OCL para a linguagem SQL-DML, e utilizando funções armazenadas, permitir que regras de negócio sejam definidas no nível DML e não apenas em DDL.

Como trabalho futuro, que já está em andamento, será implementado o mapeamento de regras de OCL para a linguagem Java (BRIAND, 2004), pois muitos SI podem exigir a validação de regras sem a invocação de funções no SGBD. Além disso, será preciso validar o mecanismo aqui proposto no SI alvo e, posteriormente, em outros SI, de forma a garantir a generalidade da solução.

4. Agradecimentos

Os autores agradecem o apoio financeiro do CNPq para a realização do projeto do SI, e da CAPES pelo suporte financeiro concedido para a bolsista.

5. Referências Bibliográficas

- BRIAND, L. C.; DZIDEK, W.; LABICHE, Y. Using Aspect-Oriented Programming to Instrument OCL Contracts in JAVA. Relatório Técnico: TR SCE-04-03. Carleton University, 2004.
- CORREA, A.; WERNER, C. Applying Refactoring Techniques to UML/OCL Models. In: UML'2004 - THE UNIFIED MODELING LANGUAGE CONFERENCE, 7., 2004.
- DEMUTH, B.; HUSSMANN, H. Using UML/OCL Constraints for Relational Database Design. In: UML'1999 - THE UNIFIED MODELING LANGUAGE CONFERENCE, 2., 1999, Fort Collins, CO, USA.
- DEMUTH, B.; HUSSMANN, H.; LOECHER, S. OCL as a Specification Language for Business Rules in Database Applications. In: UML'2001 - THE UNIFIED MODELING LANGUAGE CONFERENCE, 4., 2001, Toronto, Ontario, Canada.
- FINGER, F. Design and Implementation of a Modular OCL Compiler. 2000. Tese (Doutorado em Ciência da Computação) - Department of Computer Science Software Engineering Group, Dresden University of Technology, 2000.
- GAGNON, E. SableCC – An Object-Oriented Compiler Framework. 1998. Dissertação (Mestrado em Ciência da Computação) - School of Computer Science, McGill University, Montreal, 1998.
- GROUP, T. B. R. Guide Business Rules Project: Defining Business Rules - What Are They Really?. rev.1.3, July 2000. Disponível em: <<http://www.businessrulesgroup.org>>. Acesso em: abr.2006.
- HUSSMANN, H.; DEMUTH, B.; FINGER, F. Modular Architecture for a Toolset Supporting OCL. In: UML'2000 - THE UNIFIED MODELING LANGUAGE CONFERENCE, 3., 2000.
- MELTON, J. ISO-ANSI Working Draft – SQL. 2003.
- WARMER, J.; KLEPPE, A. The Object Constraint Language: Getting Your Models Reading for MDA. 2nd Edition, Addison Wesley, 2003.
- ZIMBRÃO, G. et al. ATENAS: Um Sistema Gerenciador de Regras de Negócio. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 15., 2001, Rio de Janeiro. Anais...
- ZIMBRÃO, G. et al. Enforcement of Business Rules in Relational Databases Using Constraints. In: SIMPÓSIO BRASILEIRO DE BANCOS DE DADOS, 18., 2003. Anais...