

ATENAS: Um Sistema Gerenciador de Regras de Negócio

Geraldo Zimbrão da Silva (IM/UFRJ)
Victor Teixeira de Almeida (COPPE/UFRJ)
Jano Moreira de Souza (COPPE/UFRJ)
Alberto Sulaiman (COPPE/UFRJ)
Francisco Gonçalves Pereira Neto (DAbM/Marinha)
{zimbrao,valmeida,jano,sulaiman}@cos.ufrj.br
COPPE /UFRJ e Marinha do Brasil.

1. Introdução

A separação das regras de negócio nos sistemas de informação apresenta uma série de vantagens amplamente aceitas na comunidade de sistemas de informação [R97, R98, D00]. No entanto, ferramentas que suportem este tipo de abordagem ainda se encontram em sua fase embrionária, apresentando uma série de limitações que comprometem o seu uso em sistemas reais, em particular em sistemas que já estejam em produção e manutenção. Este artigo apresenta uma ferramenta, a Atenas, que está sendo desenvolvida pela COPPE em conjunto com a Diretoria de Abastecimento da Marinha do Brasil. O propósito inicial desta ferramenta é representar, utilizando uma base de conhecimento formal, as regras de negócio do SINGRA – Sistema de Informações Gerenciais de Abastecimento – o sistema que controla o abastecimento da Marinha.

Segundo [G97] uma regra de negócio pode ser definida segundo duas perspectivas: a perspectiva do negócio e a perspectiva dos sistemas de informação (SI). Segundo a perspectiva do negócio, uma regra de negócio é uma diretiva destinada a influenciar ou guiar o comportamento do negócio, como suporte à política de negócio que é formulada em resposta a uma oportunidade. Segundo a perspectiva dos sistemas de informação, uma regra de negócio é uma sentença que define ou restringe algum aspecto do negócio. Pretende-se garantir a estrutura do negócio ou controlar a influência do comportamento do mesmo.

Ainda, segundo [D00], as regras de negócio são a solução para o problema da necessidade de se escrever código de forma procedural, podendo-se especificar sistemas apenas de forma declarativa. As regras de negócio, segundo ele, nos permitem automatizar o processamento de negócios.

2. Motivação

Como dito em [G97] os métodos tradicionais de análise de sistemas por muito tempo tenderam a negligenciar regras de negócio inerentes aos empreendimentos, preocupavam-se principalmente em modelar a estruturação dos dados utilizados e os processos executados pelos empreendimentos. Os métodos mais modernos de modelagem de sistemas, principalmente os que dão enfoque a orientação a objetos, já tentam lidar e resolver tais problemas. Isto pode ser visto na proposta de extensão a UML[BRJ97] chamada OCL (Object Constraint Language) em [OCL97] para dar suporte a restrições. Date em [D00] vai mais além dizendo que poderíamos até eliminar o processo de codificação de software simplesmente utilizando uma modelagem de sistemas baseado em regras de negócio com sua frase de impacto “declarative is better the procedural” que significa que é melhor que se defina sistemas apenas de forma declarativa e não procedural como é feito hoje em dia. Uma

forma de se conseguir, ao menos em parte, este objetivo, é a formalização das regras de negócio em uma base de conhecimento.

Uma dificuldade na administração das regras de negócios está no fato de que as bases de conhecimento não devem conter incongruências lógicas. Formalizar as regras de negócio utilizando-se uma linguagem declarativa permite que as mesmas sejam convertidas para uma representação em lógica de primeira ordem, o que irá facilmente permitir que tais incongruências sejam identificadas. Mais do que isso, ao se modificar uma regra, ou acrescentar uma nova, a consistência da base poderá ser testada para a verificação de incongruências, de modo a listar as regras conflitantes auxiliando a equipe no trabalho de resolver tais problemas. Além disso, a formalização de parte do conhecimento obtido durante a fase de análise é um grande passo na direção de se gerar código automaticamente.

3. Funcionalidades Especificadas

A idéia básica da ferramenta é servir como uma base formal para a documentação das regras de negócio de um sistema de informação. Cada regra de negócio é estabelecida em uma linguagem formal, seguindo o modelo apresentado em [G97]. No entanto, a ferramenta está sendo desenvolvida de forma a poder ser introduzida gradualmente em um sistema em desenvolvimento ou mesmo já em produção, como é o caso do SINGRA. Para tanto, a especificação de uma regra de negócio poderá ser feita utilizando-se uma linguagem formal (OCL, Pascal, PL/SQL), ou de maneira informal, através de um texto.

As regras de negócio estão divididas em três grupos: regras extraídas automaticamente do esquema do banco de dados, regras cadastradas manualmente pelos usuários e regras inferidas automaticamente a partir das outras duas. A ferramenta possui funcionalidade para lidar com estes três tipos de regras. Segundo o propósito original da ferramenta, que é gerenciar uma base de conhecimento sobre regras de negócio de um sistema atualmente em produção, as seguintes funcionalidades são oferecidas:

Validar o Sistema – uma vez que todas as regras de negócio estejam estabelecidas em uma linguagem formal, é possível gerar a lista de eventos de sistema (inserções, alterações etc) onde as mesmas devem ser avaliadas. De posse destas informações, é possível inspecionar o código para verificar se as regras de negócio estão realmente sendo avaliadas, e se o estão da forma correta.

Validar os dados legados – uma tarefa árdua quando um sistema é implantado em substituição a outro é a importação dos dados legados pelo sistema antigo. Mais do que colocar a base sob um novo formato, é imprescindível também assegurar a integridade dos dados, ou seja, a qualidade e acurácia da base legada. Porém, quando uma base de dados legados está para ser migrada nem sempre é viável entrar os dados simulando o uso normal do novo sistema. Em geral, será feita uma carga massiva de dados, e somente então as regras de negócio serão avaliadas. Note que esta abordagem, conquanto funcione bem para as regras simples, como as restrições de integridade, nem sempre irá funcionar para as regras de negócio mais complexas, em particular as regras que envolvem memórias de cálculos ou estados passados que não eram mantidos pelo sistema anterior. Portanto, se as regras de negócio não estiverem formalizadas, esta tarefa irá demandar a codificação de procedimentos especiais para verificar a validade das regras de negócio, e que serão utilizados apenas para a importação dos dados e descartados após isto. Com a abordagem proposta pela nossa ferramenta, será possível gerar estes procedimentos automaticamente para todas as regras que estejam formalmente estabelecidas.

Auxílio para a Extração das Regras de Negócio – segundo [R97, R98], em um sistema que estruture as regras de negócio de forma independente, as mensagens de erro são as próprias regras de negócio. Esta afirmação tem uma consequência importante para a extração das regras de negócio de um sistema já desenvolvido: os lugares do sistema onde mensagens de erro são geradas certamente contém algum tipo de regra de negócio. Certamente esta abordagem não irá revelar todas as regras de negócio, pois nem todas as regras geram mensagens de erro – em particular, regras que são derivações usualmente geram apenas resultados. No entanto, boa parte das regras de restrição podem ser obtidas desta forma – algumas até automaticamente, como é o caso dos restrições de integridade codificadas no banco de dados. Dessa forma, o esquema do banco de dados é uma fonte importante que pode ser utilizada para a extração automática de regras de negócio já formalizadas. Outra fonte importante é a documentação informal estabelecida durante a fase de análise.

Manutenção do Sistema – a ferramenta permite analisar o impacto de mudanças em regras de negócio, através do relatório de impacto de uma mudança em alguma regra de negócio. Cada alteração proposta de uma regra (introdução, remoção ou alteração) irá gerar uma lista de eventos, mapeados para trechos de código, que devem ser inspecionados de forma a garantir que as alterações necessárias sejam realizadas. Isto pode ser visto no fluxograma da figura 1. Além disso, para cada regra será mantida uma informação de grande utilidade para a manutenção do sistema: os trechos do código fonte onde a regra está sendo avaliada.

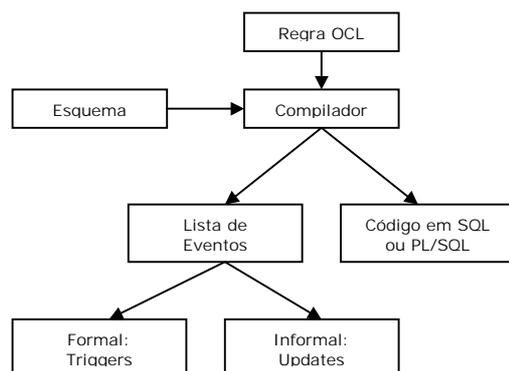


Figura 1: Esquema de avaliação do impacto de alterações na base de dados.

Além disso, a ferramenta oferece suporte para o desenvolvimento de sistemas utilizando especificações formais declarativas, e não procedurais, obtendo desta forma as vantagens apontadas em [D00]. Um compilador de OCL irá permitir que as regras de negócio formuladas em OCL sejam automaticamente convertidas para um código SQL equivalente que teste a validade da regra ou produza o seu efeito. Eventualmente, regras por demais complexas não poderão ser totalmente expressas em OCL, ou mesmo que o sejam, pode não ser possível gerar automaticamente código SQL eficiente para validá-las. Para poder lidar com esta situação, está previsto estender a ferramenta para gerar código procedural também, que poderá inclusive contar com o auxílio de programadores para ser feito de forma eficiente para os casos mais críticos.

4. Apresentação da Ferramenta

No cenário atual, a ferramenta será utilizada para extrair e separar do sistema de informação as regras de negócio gradualmente. No entanto, a ferramenta poderá ser utilizada no início do desenvolvimento de novos sistemas para que as regras de negócio sejam

formalizadas já na fase de análise, o que irá certamente aumentar a produtividade da equipe de desenvolvimento bem como a qualidade do sistema gerado.

A ferramenta foi implementada para uso em ambiente Windows implementada em Delphi com suporte a banco de dados ORACLE. Uma proposta de generalização para qualquer banco de dados está sendo preparada.

A arquitetura do Sistema Atenas pode ser vista na figura 2 abaixo. Segue ainda uma descrição mais detalhada de cada módulo bem como seus estágios de implementação.

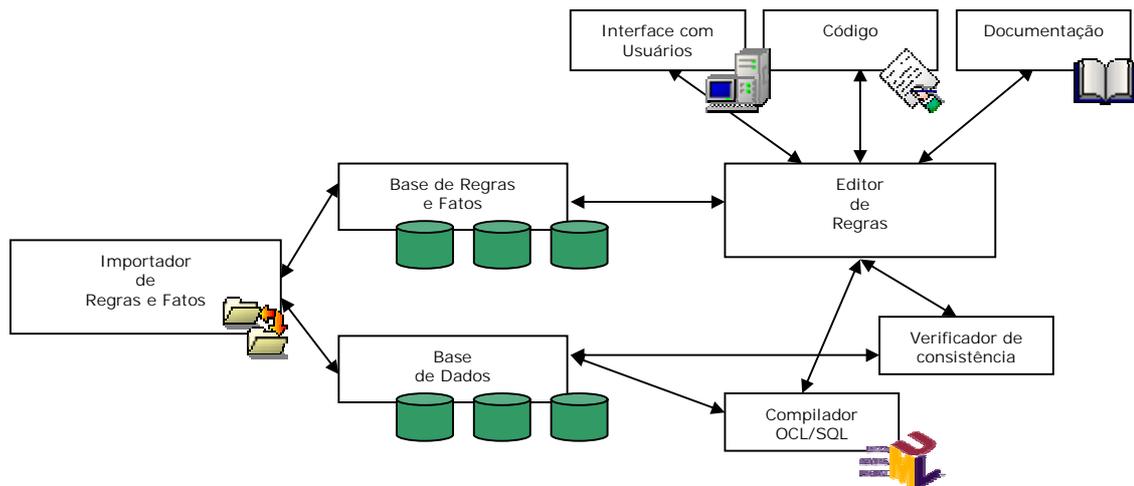


Figura 2: Arquitetura do Sistema Atenas de Regras de Negócio

4.1 Módulos implementados ou em testes:

- 1) Base de Regras e Fatos: o modelo da base de dados para o armazenamento das regras e fatos pode ser visto na figura 3.
- 2) Extrator automático de regras: extrai do esquema da base de dados informações sobre regras de negócio já formalizadas em SQL, tais como restrições de integridade, relacionamentos etc. Atualmente implementada.

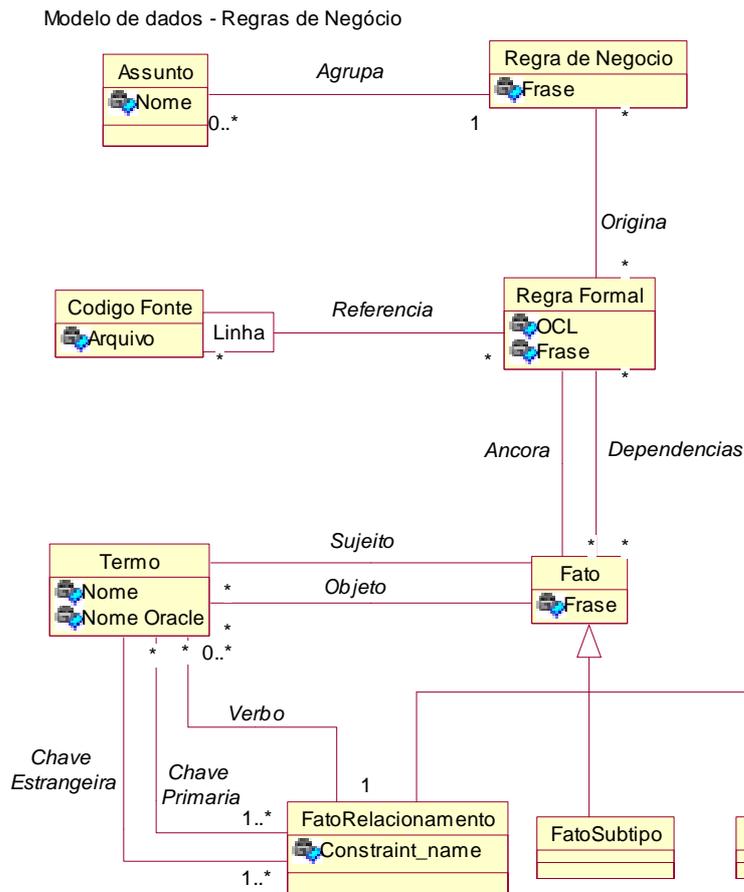


Figura 3: Modelo da Base de Regras e Fatos.

- 3) Editor de regras de negócio: oferece suporte para a entrada de regras de negócio pela equipe de analistas. As regras podem ser representadas formalmente, em diversas linguagens, ou informalmente. Atualmente, para as regras representadas formalmente, apenas um subconjunto da linguagem OCL é utilizado para gerar automaticamente a lista de eventos relevantes. Para as regras representadas informalmente, os eventos relevantes devem ser estabelecidos manualmente. Gradualmente, todas as regras devem ser formalizadas em OCL. Além disso, o editor de regras permite que as regras sejam agrupadas por assunto, e permite também a criação de referências para o código fonte que avalia a regra em questão.
- 4) Gerador da lista de Eventos X Regras: este módulo relaciona todos os eventos em que determinada regra deve ser avaliada, gerando o código SQL (a partir da especificação da regra em OCL) para tanto. Este módulo está sofrendo melhorias para eventualmente gerar código procedural (em PL/SQL), permitindo um desempenho melhor para regras mais complexas.
- 5) Verificador de Consistência (em fase de especificação): irá validar a base de regras para identificar e reportar incongruências lógicas.

4.2 Módulos em implementação:

- 6) Gerador de Regras para a validação de Dados Legados.
- 7) Analisador de Impacto de Mudanças: irá reportar os eventos, e com isto os trechos do código, que devem ser alterados para refletir mudanças nas regras de negócio.

Bibliografia

- [D00] Date, C. What not How, The Business Rules Approach to Application Development, Addison-Wesley, Reading Massachusetts, 2000.
- [R97] Ross, R. The Business Rules Book: Classifying, Defining, and Modeling Rules 2nd edition, Business Rule Solutions Inc., Houston Texas, 1997.
- [R98] Ross, R. Business Rules Concepts, The New Mechanics of Business Information Systems, Business Rule Solutions Inc., Houston Texas, 1998.
- [G97] GUIDE Business Rules Project: *Final Report*, revision 1.2, October 1997.
- [BRJ97] Booch, G.; Rumbaugh, J.; Jacobson, I.; “The Unified Modeling Language for Object-Oriented Development”, Documentation Set Version, janeiro, 1997.
- [OCL97] Object Management Group, “Object Constraint Language Specification”, 1 September 1997.