

Reengenharia Orientada a Objetos de Sistemas COBOL Utilizando Padrões de Projeto e Servlets

*Valter Vieira de Camargo**
Rosângela Ap. D. Penteado

Universidade Federal de São Carlos – UFSCar
Departamento de Computação
Caixa Postal 676
13565-905 – São Carlos – SP- Brasil

{valter, rosangel}@dc.ufscar.br

Resumo

A constante preocupação com a manutenibilidade de sistemas legados tem conduzido a diversas pesquisas. Tais sistemas, quando implementados em COBOL despertam especial atenção por parte dos engenheiros de software. Na maioria das vezes, esses sistemas devem passar por um processo de reengenharia em que tanto o paradigma de desenvolvimento como a linguagem de programação devem ser alterados. A engenharia reversa, primeiro passo de um processo de reengenharia, pode ser realizada com o apoio da abordagem Fusion/RE. O interesse na aplicação de resultados eficazes, comprovados por outros pesquisadores, como os Padrões de Projeto de software aparecem como candidatos à integração com o processo de reengenharia, principalmente, quando bancos de dados relacionais são utilizados para a persistência de dados e a linguagem de implementação é orientada a objetos. A vantagem da Internet é comprovada e a tendência de disponibilizar sistemas por meio dela é crescente. Assim, este trabalho tem interesse na criação e validação de um processo de reengenharia orientada a objetos para sistemas implementados em COBOL, alterando a linguagem de implementação para orientada a objetos, Java, utilizando Padrões e *servlets*, sem a alteração da sua funcionalidade.

Palavras-chave: Orientação a Objetos, *Persistence Layer*, Java, *servlet*, Internet, *SyBase*.

* Trabalho realizado com o apoio financeiro da CAPES.

1 – Introdução

Qualquer atividade que torne os sistemas legados mais manuteníveis e evolutíveis é de extrema relevância. Uma dessas atividades é a engenharia reversa, responsável pela recuperação de um modelo em nível de abstração mais alto que o código fonte. Se além desse modelo houver o interesse na mudança da linguagem de programação, o processo de reengenharia deve ser aplicado. Sistemas implementados em COBOL fazem parte do rol de sistemas legados.

Atualmente, surge nas empresas a necessidade de disponibilizar sistemas via Internet, seja para a venda de seus produtos ou para o acesso de seus funcionários a um banco de dados central. Tecnologias que tornem esse acesso possível são de extrema relevância.

A utilização de soluções já discutidas e viabilizadas por outros pesquisadores também auxilia no processo de reengenharia. Dessa forma, Padrões de Projeto de Software [11] são relevantes para o desenvolvimento de sistemas, já que, padronizam e diminuem o tempo de desenvolvimento.

2 – Projeto Proposto

Este trabalho visa submeter sistemas implementados em COBOL a um processo de reengenharia, mudando tanto o paradigma de desenvolvimento como a linguagem de implementação de procedimental para orientado a objetos. A abordagem Fusion/RE [3] para a realização da engenharia reversa orientada a objetos, o Padrão de Projeto *Persistence Layer* [4] para tratar as dificuldades de compatibilização entre o paradigma relacional e o orientado a objetos e *servlets* [5] para que o sistema possa ser disponibilizado via Internet, são utilizados. Dessa forma pretende-se: a) testar a abordagem Fusion/RE para sistemas implementados em COBOL e b) integrar tecnologias e abordagens diferentes como *Servlets*, Fusion/RE, Padrões de Projeto e Linguagens Orientadas a Objetos para a realização da reengenharia.

Um sistema legado real implementado em COBOL é utilizado como estudo de caso para o desenvolvimento deste projeto. A figura 1 ilustra como ocorrerá o desenvolvimento do projeto proposto. A abordagem Fusion/RE é aplicada ao sistema COBOL para se obter o modelo de análise orientado a objetos do sistema, que será utilizado no processo de reengenharia, juntamente com Padrões de Projeto, Linguagem Java e *Servlets*. O produto resultante da reengenharia será um sistema totalmente orientado a objetos e passível de ser disponibilizado via Internet.

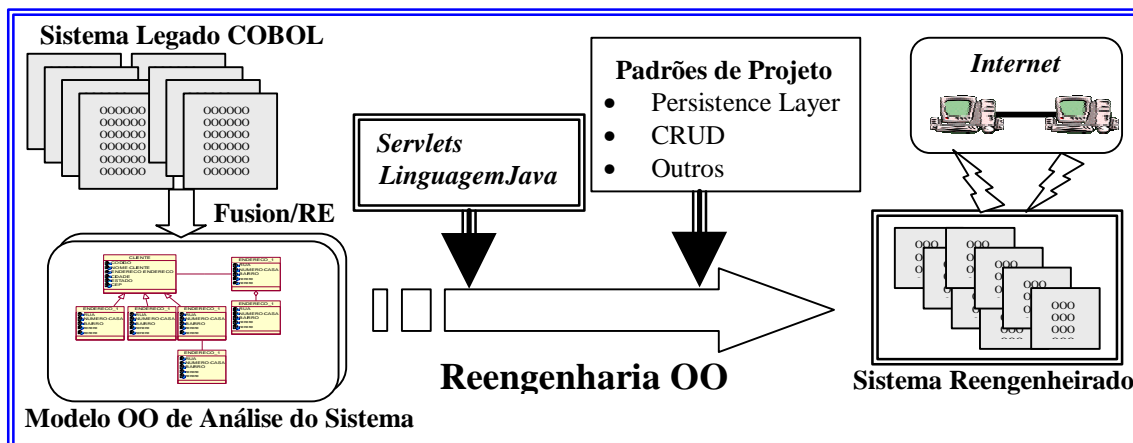


Figura 1 – Processo de Reengenharia OO com Padrões de Projeto e Java Servlets

3 – Metodologia Utilizada

3.1 – Engenharia Reversa com Fusion/RE

A abordagem de engenharia reversa utilizada neste trabalho é o Fusion/RE [3], pois, como se visa um processo de reengenharia orientada a objetos deve-se utilizar uma abordagem de engenharia reversa com o mesmo paradigma. Fusion/RE já foi utilizado em sistemas implementados em Clipper [4] e C [9], e está sendo aplicado em um sistema implementado em COBOL. A abordagem Fusion/RE possui 4 passos, que são: a) Revitalizar a Arquitetura do Sistema com Base na Documentação Existente, b) Recuperar o Modelo de Análise do Sistema Atual - (MASA), c) Abstrair o Modelo de Análise do Sistema Atual - (MAS), e c) Mapear o Modelo de Análise do Sistema para o Modelo de Análise do Sistema Atual. Essa abordagem baseia-se no método Fusion, mas neste trabalho a notação UML será utilizada para a especificação do sistema, por dispor de melhores recursos computacionais e sem perdas de semântica devido a semelhança de conceitos empregados.

3.2 – Reengenharia

O modelo de análise orientado a objetos obtido após o processo de engenharia reversa será utilizado no passo seguinte.

Jacobson e Lindström [2] descrevem três cenários para a realização da reengenharia orientada a objetos em um sistema implementado sem a utilização desse paradigma: a) troca total da linguagem de implementação sem troca da funcionalidade; b) troca parcial da linguagem de implementação sem troca da funcionalidade e, c) troca da funcionalidade. Este trabalho utilizará o primeiro cenário proposto pelos autores, pois não há interesse na mudança da funcionalidade do sistema, mas sim da linguagem de implementação, de COBOL para Java.

Reengenharia de software é definida por Jacobson e Lindström [2] como engenharia reversa + Δ + engenharia avante, sendo Δ uma modificação que o sistema pode sofrer. A partir dos modelos obtidos no passo anterior os modelos de projeto são elaborados para que a implementação do sistema ocorra. A linguagem Java é a escolhida para a implementação do sistema. Como o banco de dados é relacional, o padrão de projeto Persistence Layer [12] é utilizado para amenizar as incompatibilidades entre o paradigma orientado a objetos e o banco de dados relacional. Padrões de Projeto [11] tem por objetivo auxiliar os engenheiros de software na solução de problemas recorrentes encontrados no desenvolvimento de software. O padrão Persistence Layer [12] foi utilizado em um processo de reengenharia realizado por Cagnin [4].

Uma experiência com o desenvolvimento de um sistema simples utilizando o padrão *Persistence Layer* juntamente com a linguagem Java e *servlets* foi realizada por Camargo e outros [10]. Observou-se que a manutenção do sistema é facilitada como também a portabilidade para a aplicação.

3.3 – Java Servlets

Como este trabalho tem interesse na disponibilização de sistemas via Internet, deve-se escolher uma tecnologia recente para alcançar tal objetivo. Desse modo, optou-se pela utilização das classes *servlets* [8] pertencentes à linguagem Java [5,6]. Essa linguagem, é orientada a objetos, portátil, *multithread*, segura, fácil de aprender e permite o acesso a banco de dados via Internet. Em um ambiente de sistemas distribuídos, como é o caso da Internet, as classes Java servlet permitem tratar requisições e respostas dos/aos clientes ligados à rede.

Bergesten [8] e Brokwood [7] afirmam que quando se utiliza *scripts* CGI (Common Gateway Interface), um novo processo é criado a cada nova requisição do cliente, fazendo com que um servidor, que seja precário em recursos, se torne extremamente lento. Já no caso de servlets, uma única instância é criada e cada requisição é tratada como uma *Thread*

separada. A preferência em utilizá-lo em vez de scripts CGI é indicada sempre que se visa ganho de performance.

4 – Conclusão

A engenharia reversa do sistema do caso de estudo está sendo realizada com o apoio de uma ferramenta para que o completo entendimento do sistema ocorra. Pretende-se elaborar diretrizes que possam ser seguidas por engenheiros de software que têm programas implementados em COBOL e que desejam reengenheirá-los para a linguagens orientadas a objetos, Java, utilizando Padrões de Projeto e servlets.

Referências Bibliográficas

- [1] - **Chikovsky, E.** - Reverse Engineering and Design Recovery - A Taxonomy. IEEE Software. v. 7, n. 1, p. 13-17, 1990.
- [2] - **Jacobson, I.; Lindström, F.** - Re-engineering of Old Systems to an Object-Oriented Architecture. Proceedings of the OOSPLA'91, ACM, p.340-350, 1991.
- [3] - **Penteado, R. A. D.** – Um Método para Engenharia Reversa Orientada a Objetos. Tese de Doutorado – Instituto de Física de São Carlos, Universidade de São Paulo. São Carlos, 1996.
- [4] - **Cagnin, M. I.; Penteado, R. A. D.; Germano, F. R. S.; Masiero, P. C.** – Reengenharia com Uso de Padrões de Projeto. XIII Simpósio Brasileiro de Engenharia de Software, pág. 273-288, Outubro, 1999.
- [5] - Developing Servlets. URL : http://www.borland.com/techpubs/jbuilder/jbuilder3/distributed/serv_servlets.html.
- [6] - **Saleh, K.** - Object model in Java: Elements and Application. Elsevier Science-Information and Software Technology , p. 235-241, 1999.
- [7] - **Brockwood , T.**- Java – URL: <http://webdevelopersjournal.com/articles/wijava.html>
- [8] - **Bergsten, H.** – An Introduction to Java Servlets - URL: http://webdevelopersjournal.com/articles/intro_to_servlets.html.
- [9] - **Penteado, R.; Masiero, P.C.; Prado, A.F.; Braga, R.T.V.** – Reengineering of Legacy Systems Based on Transformation Using the Object – Oriented Paradigm. In 5th Working Conference on Reverse Engineering, Honolulu, Hawai – USA, pg. 144-153, 1998.
- [10] - **Camargo, V.V.; Prieto, A.G.; Penteado, R.D.** - Uma Experiência na Integração de UML, Padrão de Projeto e Servlet.- Aceito para apresentação no CLEI 2000, Mexico, setembro 2000.
- [11] - **Christopher Alexander: An Introduction for Object-Oriented Designers** – URL: <http://gee.cs.oswego.edu/dl/ca/ca.html>.
- [12] - **Yoder, J. W.; Johnson, R. E.; Wilson, Q. D.** – Connection Busines Objects to Relational Databases. In: Conference on the Patern Languages of Programs, 5, Monticello-IL, EUA. Proceedings, 1998.