

O PROBLEMA DE RESTAURAÇÃO DE CADASTROS

Aluno: Paulo Ivson Netto Santos , Caio Dias Valentin

Orientador: Eduardo Sany Laber

Introdução.

O cadastro de endereços de clientes é um ativo muito importante da maioria das grandes empresas da atualidade, especialmente no setor de serviços. Tal cadastro permite a empresa se comunicar com seus clientes e facilita o comércio ao permitir o envio de produtos e cobranças.

Infelizmente, muitos destes cadastros contêm uma quantidade grande erros. Estes ocorrem por diversos motivos: alguns são provenientes da digitação, outros provêm de uma ortografia incorreta. Frequentemente, o próprio cliente desconhece parte de seu endereço, escrevendo o bairro ou o CEP errado, por exemplo.

Tais erros causam prejuízos grandes a empresas, devido ao custo associado ao envio de correspondência para endereços equivocados.

Como fruto dos esforços anteriores, foi desenvolvido um software capaz de corrigir automaticamente uma base de endereços incorretos de forma eficaz e eficiente. Os resultados obtidos validaram os algoritmos implementados e comprovaram a estratégia de solução desenvolvida.

Dando continuidade aos estudos efetuados, foram observados problemas e limitações no software desenvolvido. Estes por sua vez obstruíram o estudo de novos algoritmos, já que prejudicavam a implementação e conseqüente experimentação de novas propostas.

Objetivos

- Verificar a correção do software desenvolvido anteriormente
- Elaborar uma nova proposta para uma plataforma de desenvolvimento de algoritmos que visam atacar o problema de restauração de cadastros
- Implementar as soluções propostas e validar seus resultados com base nos resultados já comprovados do software anterior

Metodologia.

A necessidade de implementar e testar novos algoritmos apontou para um problema que ainda não havia sido constatado: a extensibilidade do programa. Em outras palavras, a maneira com que o software foi construído estava dificultando a implementação de novas técnicas resultantes de nossos estudos.

Estava sendo observado um fenômeno comumente conhecido na Engenharia de Software por “*code bloat*”, onde o código de um programa se torna exponencialmente difícil de manter e de se adicionar novas funcionalidades. Vários fatores contribuem para isto, dentre os principais: a linguagem de programação utilizada, o paradigma de programação adotado, a coordenação entre os programadores, etc.

Tendo em vista o objetivo principal de estudo e pesquisa do projeto, chegou-se à conclusão de que seria necessário uma reestruturação – *refactoring* – do software. Desse

modo, seria possível dar continuidade a futuros estudos sem os empecilhos advindos das dificuldades mencionadas.

Portanto, os esforços foram voltados para uma análise dos resultados do programa antigo, como forma de identificar seus pontos fracos onde pudessem ser efetuadas melhorias. Além disso, estes resultados serviriam como base de validação para resultados futuros. Em paralelo, foram analisados os principais algoritmos implementados no programa com o mesmo intuito de identificar suas vantagens e limitações. Juntamente com a engenharia reversa do sistema e de suas estruturas, foi elaborada uma documentação do software desenvolvido.

Esta documentação serviu tanto como base para identificar as limitações na abordagem adotada quanto como ponto de partida para o desenvolvimento de novas melhorias. Assim, procurou-se eliminar os problemas identificados e ao mesmo tempo mudar o paradigma de programação adotado, de forma a evitar uma situação semelhante no futuro.

Para isto, foram adotados conceitos de programação orientada a objetos, que se revelaram melhores no sentido de atender aos requisitos do problema de restauração de cadastros. A nova versão do software foi desenvolvida em C++ de modo que foi possível maximizar o reuso de código anterior (em C), preservar seu desempenho de processamento e por fim reestruturar o sistema de forma organizada e acima de tudo eliminando as limitações encontradas.

Conclusões

Através da documentação obtida a partir da análise de resultados anteriores e da engenharia reversa do software anterior, foi desenvolvido um segundo protótipo do programa de correção automática de endereços. Este se encontra atualmente em fase final de testes para validação de seus resultados, que até agora provaram ser no mínimo promissores. A maneira como foi desenvolvida esta nova ferramenta a torna importante aliada no estudo e pesquisa de novas soluções para o problema de restauração de cadastros e os paradigmas a ele associados.

Referências

- 1 - STROUSTRUP, Bjarne: **The C++ Programming Language (Special Edition)**. Addison Wesley, 2000
- 2 - BECK, Kent; ANDRES, Cynthia. **Extreme Programming Explained : Embrace Change (2nd Edition)**. Addison Wesley, 2004
- 3 - MEYERS, Scott. **Effective C++ (3rd Edition)**. Addison Wesley, 2005
- 4 - MEYERS, Scott. **More Effective C++**. Addison Wesley, 2005
- 5 - FOWLER, Martin; BECK, Kent; BRANT, John; OPDYKE, William; ROBERTS, Don. **Refactoring: Improving the Design of Existing Code**. Addison Wesley, 1999
- 6 - GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison Wesley, 1995