

Incorporação de novas refatorações para linguagem Fortran no IDE Eclipse

Gustavo Riseti¹, Andrea S. Charão¹, Bruno B. Boniati¹

¹Universidade Federal de Santa Maria (UFSM)
{rissetti, andrea}@inf.ufsm.br, brunoboniati@gmail.com

Resumo. Refatoração é uma técnica de engenharia de software que visa aplicar mudanças internas no código fonte de aplicações, sem influenciar seus resultados. Na computação de alto desempenho, onde existem muitos códigos legados, a refatoração é pouco explorada, pois tais códigos são escritos em linguagens não orientadas a objetos, como Fortran. Este trabalho explora tal lacuna através do desenvolvimento de novas refatorações para o ambiente integrado de desenvolvimento Eclipse, utilizando-se do framework do Photran (um plugin para programação Fortran integrado ao Eclipse). A validação dessas refatorações é feita medindo-se seu impacto no desempenho de uma aplicação existente.

1. Introdução

Considerando-se o histórico das Tecnologias da Informação hoje utilizadas, nos deparamos com códigos empregados em grandes centros que tipicamente têm décadas de desenvolvimento, sendo compostos por centenas de milhares de linhas de código e programados por diversas equipes diferentes, cada uma utilizando alguma técnica de programação e costumes diferentes. Um efeito direto deste histórico é o uso de linguagens de programação muito antigas, que possuem estilos e técnicas de programação que são consideradas perigosas, dificultando a leitura de código e a identificação de possíveis erros de programação, assim como a manutenção do código, que é fundamental durante todo o ciclo de vida de um *software*. Uma maneira de contornar tais problemas consiste no uso de técnicas de refatoração de código fonte.

Refatoração é uma técnica de engenharia de *software* que compreende o processo de modificar um sistema para melhorar a estrutura interna do seu código, sem alterar o seu comportamento externo, como resultados, funcionalidades e saídas. Essa técnica está permanentemente presente no ciclo de vida de uma aplicação [Fowler 2004], e vem sendo amplamente utilizada nas linguagens de orientação a objetos mais recentes, como Java e C++. A maioria das técnicas de refatoração disponíveis hoje são destinadas ao paradigma da orientação a objetos, que possui características que facilitam o processo, sendo que linguagens mais antigas, como é o caso de Fortran, enfrentam uma forte carência no quesito de técnicas automatizadas de refatoração.

O processo de evolução é uma propriedade natural e que sempre está presente no processo de desenvolvimento de um *software*. O objetivo principal do uso da refatoração é melhorar aspectos não funcionais da aplicação tais como legibilidade, simplicidade, organização e desempenho. Embora possam ser aplicadas manualmente sobre o código fonte, o grande ganho se dá quando as técnicas de refatoração são automatizadas por meio de ferramentas que atuam sobre o código fonte previamente escrito [Roberts 1999].

O presente trabalho objetiva explorar a carência de processos de refatoração da linguagem Fortran através da automatização de técnicas de refatoração para a ferramenta Photran (um *plugin* para edição de código Fortran no IDE Eclipse), dando continuidade a um trabalho iniciado em um mestrado da computação [Boniaty 2009, Boniaty et al. 2009]. A partir da identificação de oportunidades de refatoração para código Fortran, a meta deste trabalho é de implementá-las e agregá-las ao *framework* do Photran de forma que possam ser utilizadas livremente por sua comunidade de usuários.

2. Refatoração e Fortran

A refatoração de *software* não é uma prática nova, afinal, aplicar melhorias e reestruturações ao código fonte é um processo implícito e contínuo à tarefa de desenvolvimento [Fowler 2004].

Normalmente questiona-se quando deve-se ou não utilizar refatorações em um determinado *software*. Para responder a tais questionamentos, existem alguns indicadores de necessidade de refatoração, como métodos ou classes muito extensos, métodos com excessivos parâmetros, falta de clareza ou legibilidade do código [Fowler 2004].

As técnicas de refatoração são geralmente simples, e quando aplicadas isoladamente, geralmente produzem pequenas modificações. Dependendo da melhoria que se deseja, pode ser necessário aplicar diversas técnicas de refatoração sucessivamente, obtendo-se como resultado um código fonte melhor estruturado e mais legível.

Na linguagem Fortran, a grande quantidade de códigos legados passam por processos manuais de refatoração, pois existem poucas ferramentas que auxiliam o trabalho com essa linguagem. Porém, a facilidade no uso de refatorações se dá quando as técnicas são automatizadas, permitindo a manipulação de diversos códigos fonte simultaneamente, sem ter de fazer todo o processo de forma manual, reduzindo assim o risco de erros e inconsistências, além de reduzir também o trabalho e o custo de desenvolvimento e manutenção da aplicação [Fowler 2004].

2.1. Eclipse IDE e o *plugin* Photran

O Eclipse é um IDE (*Integrated Development Environment*) de desenvolvimento que tem suporte para diversas linguagens, como C, C++, Java, e recentemente Fortran, com o uso do *plugin* Photran, que torna o Eclipse um IDE para Fortran 77, 90, 95 e 2003.

O propósito para a criação do Photran foi gerar uma ferramenta de refatoração para a linguagem Fortran, sendo que o mesmo possui um analisador sintático completo e uma representação de programa, com o Photran VPG (*Virtual Program Graph*), que gera uma AST (*Abstract Syntax Tree*) do código fonte, permitindo a sua manipulação e edição [Overbey 2009]. Assim, além de todos os recursos disponibilizados por IDEs em geral, o Photran oferece uma infraestrutura para refatoração de código fonte, com representações abstratas do programa (que permitem navegar e alterar sua estrutura em alto nível), mecanismos para visualizar e comparar diferenças antes e depois da refatoração e ainda meios de cancelar ou desfazer uma refatoração efetuada.

3. Técnicas de refatoração implementadas

Para o desenvolvimento deste trabalho foram feitas pesquisas sobre o funcionamento do *framework* do Photran, estudando-se as refatorações já disponíveis por padrão no *plu-*

gin. Estudou-se também as técnicas desenvolvidas pelo Bruno Boniati, em seu mestrado [Boniati 2009]. Obtendo-se o conhecimento necessário, passou-se a pesquisar novas oportunidades de refatoração para Fortran, apresentando-se os resultados neste trabalho.

Até o presente momento, foram implementadas e integradas ao Photran duas novas técnicas de refatoração, *Standardize Statements* e *Remove Unused Variables*. A primeira tem o objetivo de fazer uma padronização na declaração de variáveis, uma vez que a linguagem Fortran permite que variáveis sejam declaradas de diversas maneiras. Com a padronização, um código que passa por diversos programadores permanece com um mesmo padrão nas declarações, melhorando a legibilidade do código e o deixando melhor estruturado, tendo seus comandos de declarações normalizados. Na Figura 1, pode-se observar um exemplo do uso desta técnica no Photran, onde é exibido uma tela de *Preview*, mostrando as diferenças entre o código original e o código refatorado.

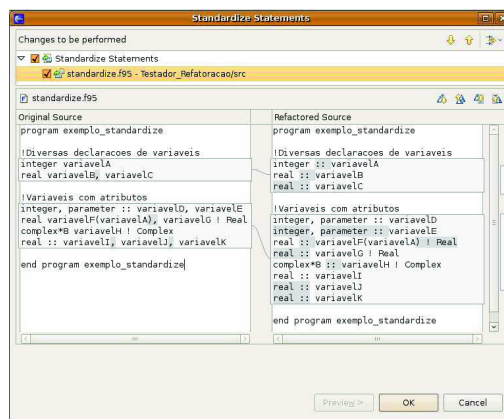


Figura 1. Exemplo da refatoração *Standardize Statements*

A segunda técnica implementada visa eliminar variáveis não utilizadas ao longo do código fonte. Na maioria das vezes em que se codifica um *software*, sobram variáveis inutilizadas ao longo do código, muitas vezes por erro do programador, que comenta um trecho do código e esquece de comentar as declarações das variáveis contidas no bloco comentado. Tendo em vista que tais erros ocorrem com grande frequência, essa técnica foi criada com a finalidade de eliminar tais variáveis esquecidas, pois dependendo do compilador usado, essa eliminação além de melhorar a legibilidade do código, pode diminuir o tamanho do executável gerado, e também é de grande valia quando se quer paralelizar o código, uma vez que ter menos variáveis facilita o controle dos acessos concorrentes na memória, podendo portanto melhorar o desempenho da aplicação.

4. Avaliação

Para avaliar o impacto do uso de técnicas de refatoração em aplicações escritas em Fortran, o trabalho utilizou-se do código fonte de uma aplicação cedida pelo Laboratório de Micrometeorologia, vinculado à Universidade Federal de Santa Maria. O objetivo do

estudo de caso é avaliar de que forma o uso das técnicas aqui implementadas podem influenciar no desempenho (de forma positiva ou negativa) da referida aplicação.

A aplicação testada foi escrita em linguagem Fortran 77, tendo grande importância para pesquisas no laboratório, sendo utilizada para analisar conjuntos numerosos de dados captados por sensores em estações meteorológicas. Dado seu tempo de vida e algumas construções obsoletas utilizadas no seu desenvolvimento, apresenta uma alta complexidade de compreensão e manutenção. O compilador usado para os testes foi o *gfortran*.

O tempo médio de execução do código original com um arquivo de entrada de 106.8 MBytes, em um computador Intel Core 2 Duo 1.8 GHz foi de 21m32.184s, com as opções de otimização do compilador desligadas. Após a análise do tempo de execução do código original, foram aplicadas as técnicas *Standardize Statements* e *Remove Unused Variables* em todas as subrotinas do *software*, resultando em um tempo médio de execução de 21m36.069s, com o mesmo arquivo de entrada e mesmas condições de compilação.

Com esses resultados pode-se concluir que com os usos das refatorações implementadas neste trabalho, o código fonte fica com melhor legibilidade e melhor estruturado, e o desempenho não é afetado negativamente, uma vez que a diferença de tempo foi de apenas 3.885s (aproximadamente 0,3% do tempo). Assim, é possível utilizar essas técnicas em qualquer código Fortran sem afetar negativamente o seu desempenho.

5. Conclusão

O presente trabalho explora a automatização e utilização de técnicas de refatoração em *softwares* escritos em Fortran, objetivando melhorar a legibilidade do código. A ferramenta Photran, um *plugin* para o Eclipse, foi utilizada como IDE uma vez que permite estender funcionalidades de refatoração e aplicá-las sobre o código Fortran.

Os resultados da avaliação demonstram que técnicas como *Standardize Statements* e *Remove Unused Variables*, que melhoram substancialmente a legibilidade de código, não influenciam negativamente o desempenho da aplicação. Dependendo da forma que são utilizadas, ao passo que tornam o código mais legível e melhor organizado, não introduzem gargalos que reduzem de forma significativa ou perceptível o desempenho.

Pode-se dizer que o Photran é uma ferramenta de grande valia para preencher a lacuna existente entre a grande quantidade de código Fortran legado e o limitado número de ferramentas de apoio ao desenvolvimento com técnicas de refatoração integradas.

Referências

- Boniati, B., Charao, A., and Stein, B. (2009). Automação de refatorações para programas fortran de alto desempenho. In *WSCAD-SSC'09: X Workshop em Sistemas Computacionais de Alto Desempenho / X Simpósio em Sistemas Computacionais*, São Paulo.
- Boniati, B. B. (2009). Refatoração de programas fortran de alto desempenho. Master's thesis, Universidade Federal de Santa Maria, Santa Maria, RS.
- Fowler, M. (2004). *Refatoração: Aperfeiçoando O Projeto De Código Existente*.
- Overbey, J. L. (2009). Virtual program graph. Disponível em: <http://jeff.overbz/software/vpg/doc/>. Acesso em: dezembro de 2009.
- Roberts, D. B. (1999). *Practical analysis for refactoring*.