Catálogo de Refatorações para a Evolução de Programas em Linguagem Fortran

Gustavo Rissetti[1], **Andrea S. Charão**[1], **Eduardo K. Piveta**[1]

[1] Universidade Federal de Santa Maria (UFSM)
{rissetti,andrea,piveta}@inf.ufsm.br

1. Introdução

Refatoração é o processo de modificar um sistema de software para melhorar sua estrutura interna, sem alterar seu comportamento externo observável (resultados, funcionalidades e saídas) [Opdyke 1992, Fowler et al. 1999]. Essa técnica pode ser usada para auxiliar o processo de evolução de software, permitindo que o programador siga instruções bem definidas para modificar trechos de código e assim atender aos novos requisitos do sistema.

Este trabalho propõe um catálogo de refatorações voltadas à evolução de códigos Fortran legados. O catálogo proposto é composto por um conjunto de dez refatorações que possibilitam evoluir o padrão da linguagem de programação usada nos aplicativos de software. Algumas das refatorações do catálogo foram implementadas no *framework* Photran [Dragan-Chirila 2004, Eipe 2004], um *plugin* integrado ao Eclipse que atua sobre código Fortran e disponibiliza uma infraestrutura básica para a análise sintática de programas Fortran e a manipulação de suas árvores sintáticas.

2. Catálogo de Refatorações

A Tabela 1 mostra um resumo sobre as refatorações propostas neste catálogo, as quais almejam usar melhores construções da linguagem Fortran em determinadas circunstâncias, melhorando o projeto do código e evoluindo o padrão da linguagem usada nas aplicações.

Das dez refatorações propostas neste trabalho, seis foram implementadas no *plugin* Photran: *Extrair Subprograma para Módulo*, *Mover Subprograma para um Módulo*, *Extrair Variáveis para Tipo Derivado de Dados*, *Mover Variáveis para um Tipo Derivado de Dados*, *Converter IF-THEN-ELSE Aninhados em SELECT CASE* e *Converter laços DO em construções FORALL*. As cinco primeiras tratam da evolução de código Fortran 77 para Fortran 90, enquanto a última se refere à evolução de Fortran 90 para Fortran 95.

Para avaliar as refatorações implementadas, as mesmas foram aplicadas em aplicativos de software reais, com o objetivo de verificar seu funcionamento em códigos de diferentes versões da linguagem Fortran, que usam padrões antigos de programação, possibilitando evoluir seus códigos.

3. Conclusão

Neste trabalho, foi explorada a utilização de técnicas de refatoração sobre aplicações escritas em linguagem Fortran, objetivando melhorar o projeto de código, assim como a legibilidade do mesmo, através da evolução dos padrões da linguagem de programação utilizados em aplicações legadas. Para isso, foi proposto um catálogo de refatorações para a evolução de programas em linguagem Fortran.

Tabela 1. Resumo das refatorações propostas

Refatoração	Objetivo
Converter laços DO em Operações Matriciais Nativas	Converte laços *DO* que efetuam operações matriciais (como somas, multiplicação, inicialização, etc) em operações matriciais nativas do Fortran 90, podendo melhorar a legibilidade de código e o desempenho de execução da aplicação.
Mover Subprograma para um Módulo	Move um subprograma para um módulo existente, melhorando a organização estrutural do código e possibilitando que o subprograma seja usado em qualquer escopo da aplicação.
Extrair Subprograma para Módulo	Extrai um subprograma para um novo módulo definido pelo usuário, melhorando a organização estrutural do código e possibilitando que o subprograma seja usado em qualquer escopo da aplicação.
Mover Variáveis para um Tipo Derivado de Dados	Move variáveis para um tipo derivado de dado existente, facilitando a visualização e a manipulação de tais variáveis, servindo também como um auxilio para a refatoração *Extrair Variáveis para Tipo Derivado de Dados*.
Extrair Variáveis para Tipo Derivado de Dados	Extrai um conjunto de variáveis semelhantes, usadas em um artefato de software, para um tipo derivado de dados, facilitando a visualização e a manipulação de tais variáveis.
Converter IF-THEN-ELSE Aninhados em SELECT CASE	Converte comandos *IF-THEN-ELSE* aninhados que avaliam a igualdade de valores de uma única variável em uma estrutura de controle *SELECT CASE*, que é específica para esse tipo de operação, melhorando a legibilidade de código.
Converter laços DO em construções FORALL	Converte laços *DO* que operam sobre vetores e matrizes em construções do tipo *FORALL*, permitindo executar os laços paralelamente em uma arquitetura multi-processada, e aumentar o desempenho de execução da aplicação.
Converter Programa Procedural em Programa Orientado a Objetos	Converte o paradigma de programação usado no código-fonte, oferecendo melhores maneiras de separar o código em tarefas independentes, e permitindo construir códigos baseados em rotinas já existentes.
Converter laços DO em laços DO CONCURRENT	Converte laços *DO* convencionais em laços *DO CONCURRENT*, permitindo a execução paralela do laço quando não existem dependências de dados em seu interior, aumentando o desempenho de execução da aplicação.
Converter MPI em Coarrays	Usa o recurso de *coarrays* no lugar de bibliotecas externas que promovem paralelismo, como a MPI (*Message Passing Interface*), por exemplo. Com *coarrays* o código fica com um bom desempenho e com melhor legibilidade.

As refatorações automatizadas no Photran foram validadas em aplicações escritas em Fortran, tendo resultados satisfatórios. Com a aplicação das refatorações *Extrair Subprograma para Módulo*, *Mover Subprograma para um Módulo*, *Extrair Variáveis para Tipo Derivado de Dados*, *Mover Variáveis para um Tipo Derivado de Dados* e *Converter IF-THEN-ELSE Aninhados em SELECT CASE* foram obtidos códigos mais legíveis, adequando-os aos novos padrões da linguagem Fortran. Com a aplicação da refatoração *Converter laços DO em construções FORALL*, além de se obter um código com um padrão mais atual da linguagem, ganhou-se desempenho na execução da aplicação refatorada, pois com o uso de construções *FORALL*, é possível paralelizar os laços que envolvem operações de vetores e matrizes quando a arquitetura utilizada é multi-processada. As seis refatorações implementadas no Photran já foram enviadas ao comitê de avaliação da ferramenta para que sejam adicionadas em sua versão oficial.

Referências

Dragan-Chirila, J. (2004). Integrating a fortran 90 parser into an eclipse environment. Master's thesis, University of Illinois, Urbana-Champaign, EUA.

Eipe, R. M. (2004). Extending eclipse to create an ide plugin for a new language with fortran as a case study. Master's thesis, University of Illinois, Urbana-Champaign, EUA.

Fowler, M., Beck, K., Brant, J., Opdyke, W., and Roberts, D. (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, Boston, Massachusetts.

Opdyke, W. (1992). *Refactoring Object-Oriented Frameworks*. Tese de doutorado, University of Illinois, Urbana-Champaign, EUA.