

RESENHA CRITICA DO ARTIGO "On the Criteria To Be Used in Decomposing Systems into Modules"

Resenha da obra: PARNAS, D.L. On the Criteria To Be Used in Decomposing Systems into Modules. **Communications of the ACM**, Volume 15, Number 12, December 1972.

A presente resenha se presta a analisar o artigo "On the Criteria To Be Used in Decomposing Systems into Modules" de D.L. Parnas.

O artigo se divide em introdução, nos capítulos: *A Brief Status Report, Expected Benefits of Modular Programming, What Is Modularization?, Example System 1: A KWIC Index Production System, Modularization 1, Modularization 2, Comparison of the Two Modularizations, The Criteria, Improvement in Circular Shift Module, Efficiency and Implementation, A Decomposition Common to a Compiler and Interpreter for the Same Language, Hierarchical Structure* e conclusão.

A proposta do artigo é questionar os critérios tradicionalmente utilizados para dividir sistemas em módulos, propondo uma abordagem alternativa baseada no conceito de "information hiding" (ocultação de informação). Parnas reconhece que, embora a programação modular seja amplamente aceita como benéfica, raramente se discutem os critérios específicos para realizar essa divisão.

Nessa via, o autor identifica três benefícios esperados da programação modular: gerencial (redução do tempo de desenvolvimento através do trabalho independente de grupos), flexibilidade do produto (possibilidade de mudanças drásticas em um módulo sem afetar outros) e compreensibilidade (possibilidade de estudar o sistema módulo por módulo). Para demonstrar sua tese, Parnas utiliza um sistema de exemplo simples: um gerador de índice KWIC (Key Word In Context).

O artigo apresenta duas modularizações distintas para o mesmo sistema. A primeira segue a abordagem convencional, dividindo o sistema de acordo com as etapas de processamento sequencial:

Modularização 1 (Convencional): Módulo 1 (Input): Lê dados de entrada e os armazena; Módulo 2 (Circular Shift): Prepara índice de deslocamentos circulares; Módulo 3 (Alphabetizing): Ordena alfabeticamente os deslocamentos; Módulo 4 (Output): Produz a saída formatada; Módulo 5 (Master Control): Controla a sequência entre módulos.

Modularização 2 (Information Hiding): Módulo 1 (Line Storage): Fornece funções para armazenar e acessar linhas; Módulo 2 (Input): Lê dados originais e chama o módulo de armazenamento; Módulo 3 (Circular Shifter): Cria a impressão de um repositório contendo todos os deslocamentos circulares; Módulo 4 (Alphabetizer): Fornece funções para ordenação alfabética; Módulo 5 (Output): Produz a impressão desejada; Módulo 6 (Master Control): Similar ao anterior.

A diferença fundamental entre as abordagens reside nos critérios de decomposição. A primeira utiliza o fluxo de processamento como base - cada módulo corresponde a uma etapa

maior do processamento. A segunda emprega "information hiding" como critério - cada módulo é caracterizado pelo conhecimento de uma decisão de design que oculta de todos os outros.

Parnas demonstra a superioridade da segunda abordagem através da análise de capacidade de mudança. Na modularização convencional, alterações como formato de entrada, decisão de manter linhas em memória, empacotamento de caracteres, ou estratégia de indexação requerem modificações em múltiplos módulos. Na abordagem baseada em "information hiding", essas mesmas mudanças ficam confinadas a módulos específicos, demonstrando maior flexibilidade e manutenibilidade.

O autor também aborda questões de eficiência, reconhecendo que a segunda decomposição pode ser menos eficiente se implementada convencionalmente devido ao overhead de chamadas entre módulos. Contudo, ele sugere técnicas alternativas de implementação que preservam os benefícios conceituais sem comprometer a performance, como inserção de código pelo assembler ou transferências especializadas.

Por fim, Parnas estabelece alguns critérios específicos para decomposição baseada em "information hiding": estruturas de dados e seus procedimentos devem pertencer a um único módulo; sequências de instruções para chamar rotinas devem fazer parte do mesmo módulo da rotina; formatos de blocos de controle devem ser ocultados; códigos de caracteres e ordenações devem ser encapsulados; e sequências de processamento devem ser ocultadas sempre que possível.

O artigo representa um marco fundamental na engenharia de software, introduzindo conceitos que se tornaram pilares do desenvolvimento moderno. A proposta de "information hiding" antecipou princípios que hoje reconhecemos como encapsulamento e abstração, fundamentais na programação orientada a objetos e no design de APIs. Após 50 anos de sua publicação, os insights de Parnas permanecem extremamente relevantes, influenciando desde arquiteturas de microsserviços até práticas de desenvolvimento ágil.

A metodologia do artigo é particularmente sólida: ao utilizar um exemplo concreto e comparar duas abordagens sistematicamente, Parnas oferece evidência empírica para suas afirmações teóricas. A honestidade intelectual do autor ao reconhecer limitações e trade-offs da abordagem proposta fortalece a credibilidade do trabalho.

É importante contextualizar que algumas preocupações do artigo refletem limitações tecnológicas da época, como restrições de memória e processamento. Contudo, os princípios fundamentais transcendem essas limitações específicas. A ênfase na ocultação de decisões de design continua sendo uma estratégia crucial para lidar com a complexidade crescente dos sistemas modernos.

O artigo também demonstra notável visão prospectiva ao antecipar que decisões de design, mais do que fluxos de processamento, são os fatores determinantes para a evolução e manutenção de sistemas. Essa percepção se alinha perfeitamente com práticas contemporâneas como Domain-Driven Design e arquiteturas evolutivas, validando a atemporalidade dos conceitos apresentados.