

# Análise evolutiva do Linux através de Design Structure Matrices

MATE08 - Tópicos em Engenharia de Software 1 (2012.2)

Joenio Marques da Costa

**Resumo**—Este trabalho apresenta um relatório técnico sobre estudo exploratório do código-fonte do Linux[1] a partir de Design Structure Matrices (DSM) ao longo de 22 versões do projeto. O objetivo é demonstrar o uso de Design Structure Matrices no contexto de software bem como validar a ferramenta desenvolvida durante este trabalho. Ao longo de todo o estudo faz-se referências ao artigo "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code"[2] que deu origem a esta pesquisa e é parcialmente reproduzido aqui.

**Palavras-chave:** DSM, Analizo, Change Cost, LCOM4.

## I. INTRODUÇÃO

Muitos estudos tem abordado compreensão e visualização de softwares complexos, dentre estes alguns focam no uso de *DSM* a fim de se ter uma visão geral do relacionamento entre as entidades que compõe o sistema, além de obter meios para cálculo de métricas como *Change Cost*, uma medida que representa o custo médio de uma mudança no design de um software em termos do potencial impacto desta mudança em outras partes do sistema.

Este estudo analisa a evolução do Linux e desenvolve ferramentas para geração de *DSM* e para o cálculo da métrica *Change Cost* a partir da ferramenta Analizo<sup>1</sup>, em seguida faz um comparativo entre esta e *LCOM4*[3], bem como avalia os resultados obtidos em relação ao artigo referência[2].

## II. DESENVOLVIMENTO

22 versões do Linux foram obtidas através do site FUNET<sup>2</sup> e analisadas através da ferramenta Analizo[4], uma ferramenta para análise de código-fonte e visualização de software. O Analizo no entanto não possui suporte a *DSM* ou *Change Cost*, portanto foi necessário implementar estes novos recursos nesta ferramenta<sup>3</sup>.

Em resumo as etapas seguidas neste estudo foram as seguintes:

- 1 Implementar suporte a geração de DSM
- 2 Desenvolver algoritmo para cálculo de Change Cost
- 3 Testar e validar DSM e Change Cost
- 4 Analisar o Linux
- 5 Traçar um paralelo entre métricas
- 6 Avaliar resultados em relação ao artigo referência[2]

<sup>1</sup><http://analizo.org>

<sup>2</sup><http://www.nic.funet.fi/pub/Linux/kernel>

<sup>3</sup><http://analizo.org/Development/ActionItem34>

### A. Implementação e validação

Uma *DSM* é a representação de um grafo de dependência em forma de uma matriz quadrada, implementar uma ferramenta para este fim se resume a transformar um grafo em uma matriz e em seguida gerar uma visualização para esta matriz, isto foi implementado no Analizo a partir do módulo *Graph*<sup>4</sup>, uma implementação madura de estruturas de dados e algoritmos para grafos e do módulo *Graph::Writer::DSM*, um módulo para representação visual de um grafo em forma de *DSM*. O resultado desta implementação pode ser visto na Figura 1.

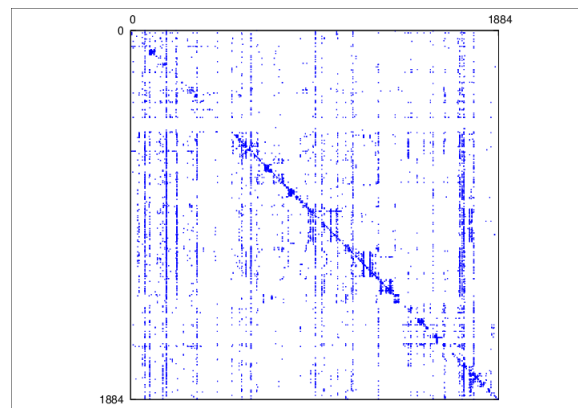


Figura 1. Design Structure Matrix do Linux 2.1.105 gerado pelo Analizo

O *Change Cost* representa o impacto que uma mudança localizada em um certo ponto pode causar em outros pontos do mesmo sistema, ele é calculado a partir de uma matrix chamada *Reachability Matrix*[5], foi implementado também através do módulo *Graph* via método *transitive\_closure\_matrix*. Por fim calcula-se o *Fan-out* de cada linha da matriz e divide-se pelo total de linhas para chegar ao resultado final de *Change Cost*.

A partir da implementação de *DSM* e *Change Cost* foram feitas várias interações para validação da implementação com base em um pequeno contendo apenas 6 arquivos/módulos.

### B. Análise do Linux

Após validação da implementação iniciou-se a análise de cada versão do Linux extraíndo para cada uma: *DSM*, *Change Cost*, *LCOM4* médio e o número de arquivos, um resumo com todos estes dados pode ser visto na Tabela I. Estes dados foram comparados aos dados do artigo referência[2] a fim de

<sup>4</sup><http://metacpan.org/module/Graph>

Tabela I  
RESUMO DA EVOLUÇÃO DO LINUX

Versão	#Arquivos	Change Cost	LCOM4
0.01	45	0.23	2.60
0.11	52	0.33	4.66
0.12	63	0.36	4.91
0.95	67	0.35	4.97
0.99.15	294	0.21	4.34
1.0	295	0.22	4.34
1.1.13	309	0.21	3.90
1.1.70	358	0.26	3.59
1.1.95	416	0.20	3.64
1.2.0	416	0.20	3.64
1.2.9	416	0.20	3.75
1.2.13	416	0.20	3.64
1.3.0	422	0.19	3.70
1.3.55	585	0.21	2.13
1.3.100	840	0.24	1.76
2.0	874	0.23	1.71
2.0.34	974	0.24	2.00
2.0.40	1082	0.25	1.65
2.1.0	877	0.24	2.09
2.1.105	1885	0.18	2.64
2.1.132	2081	0.17	2.69
2.2.0	1957	0.18	5.63

corrigir possíveis inconsistências e de se aproximar ao máximo possível dos resultados encontrados no artigo.

### C. Relação entre métricas

*Change Cost* não indica design bom ou ruim, mas sinaliza em grau de comparação que um design é mais modular que outro, ou ainda que a modularização está aumentando ou diminuindo ao longo do tempo. É possível ver na Figura 2 por exemplo que o *Change Cost* não se altera em relação ao número de arquivos do projeto.

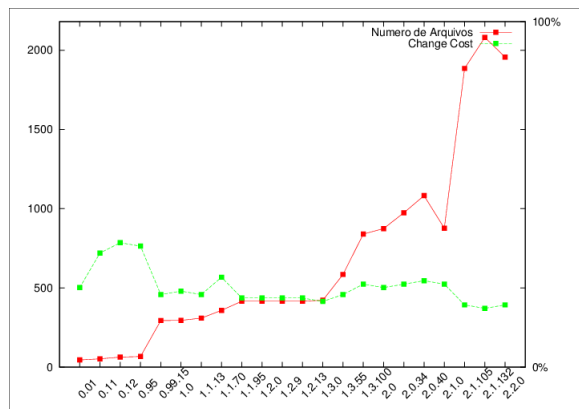


Figura 2. Relação entre Change Cost e Número de Arquivos

Assume-se que em designs menos acoplados a medida de *Change Cost* também será menor. A partir desta afirmação é possível relacionar *LCOM4* (*Lack of Cohesion in Methods*) com *Change Cost*, uma vez que coesão e acoplamento possuem relação direta. A Figura 3 mostra o quão próximos se mantém *Change Cost* e *LCOM4*, apesar de não ser possível traçar afirmações sem um estudo aprofundado, intuitivamente estas 2 medidas estão relacionadas.

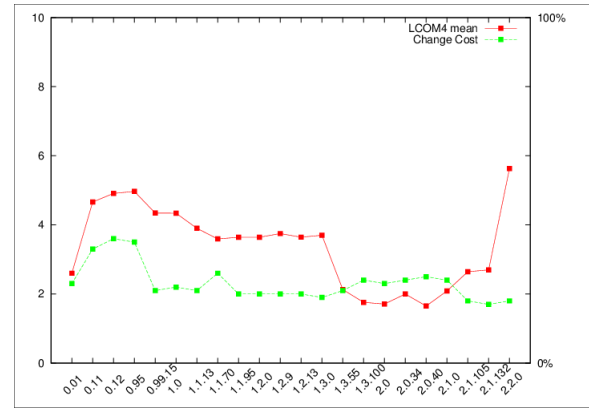


Figura 3. Relação entre Change Cost e LCOM4 médio

## III. CONCLUSÕES

Foi ilustrado brevemente como se implementou suporte a *DSM* e cálculo de *Change Cost* no *Analizo* a partir do estudo evolutivo do Linux ao longo de 22 versões. Os dados obtidos não são conclusivos uma vez que o comparativo feito com os valores encontrados no artigo de referência[2] não coincidem.

Empiricamente é possível levantar dúvidas sobre a validade dos resultados obtidos no artigo referência[2], uma vez que o autor não deixa claro quais ferramentas utilizou para gerar *DSM* ou calcular *Change Cost*. O autor calculou por exemplo que o Linux 2.1.105 possui 1678 arquivos e tem *Change Cost* = 5.16%. Ao passo que foi encontrado aqui, conforme pode ser visto na Tabela I, 1885 arquivos e *Change Cost* = 18% para esta mesma versão.

## REFERÊNCIAS

- [1] Wikipedia, "Linux (núcleo)," 2013. [Online; acessado 17 Abril de 2013].
- [2] A. M. J. R. C. Baldwin, "Exploring the structure of complex software designs an empirical study of open source and proprietary code," p. 40, 2004.
- [3] M. Hitz and B. Montazeri, "Measuring coupling and cohesion in object-oriented systems," in *Symposium on Applied Corporate Computing*, (Monterrey, Mexico), 2001.
- [4] A. T. J. C. J. M. P. M. L. R. R. L. A. C. C. F. Kon, "Analizo: an extensible multi-language source code analysis and visualization toolkit," p. 6, 2010.
- [5] A. F. H. K. Wimmer, "Reachability matrices and cyclic matrices," *Electronic Journal of Linear Algebra*, p. 8, 2010.