

Visualização de Design Structure Matrices com NodeJS + Analizo

MATE26 - Tópicos Especiais em Engenharia de Software II
(2013.1)

Joenio Marques da Costa
joenio@colivre.coop.br

Daniela Soares Feitosa
daniela@colivre.coop.br

07 de junho de 2013

Este trabalho visa o desenvolvimento de uma interface Web para visualização de *Design Structure Matrices (DSM)* com base na ferramenta *Analizo*¹, um conjunto de ferramentas para análise e visualização de código fonte extensível e independente de linguagem.

DSM é uma ferramenta que destaca a estrutura inerente do design de um produto ou projeto examinando as dependências existentes entre seus componentes através de uma matrix simétrica[1]. Ela foi inicialmente utilizada (Steward, 1981) no contexto de gestão de projetos com objetivo de melhorar o fluxo de informações entre tarefas e atividades, e tem sido utilizada constantemente com este objetivo como visto em [2]

Desde então tem sido utilizada em diversas áreas [10, 11, 14] da engenharia de software, como compreensão, visualização, análise de impacto, e outras. Uma importante característica da *DSM* para visualização de software é que é possível exibir uma grande quantidade de informações de uma forma compacta, sendo possível demonstrar o design de softwares complexos de forma bastante compacta[3].

DSMs podem ser utilizada para identificar possíveis formas de reorganizar os elementos de um sistema a fim de reduzir custos de coordenação, facilitando o fluxo de informações entre os elementos. Elementos com um alto nível de dependência são agrupados em cluster ou módulos.

Para guiar este processo uma gama de algoritmos de clusterizing tem sido desenvolvidos, com objetivo de minimizar custo de coordenação, este trabalho tem como objetivo implementar ao menos um desses algoritmos na ferramenta *Analizo*.

O *Analizo* já possui suporte a construção de *DSM* de projetos de software escritos em C, C++ ou Java, como pode ser visto na Figura 1.

Contudo a visualização de *DSM* provida pelo *Analizo* é estática e não oferece nenhuma interação com o usuário, este trabalho visa evoluir esta funcionalidade

¹<http://analizo.org>

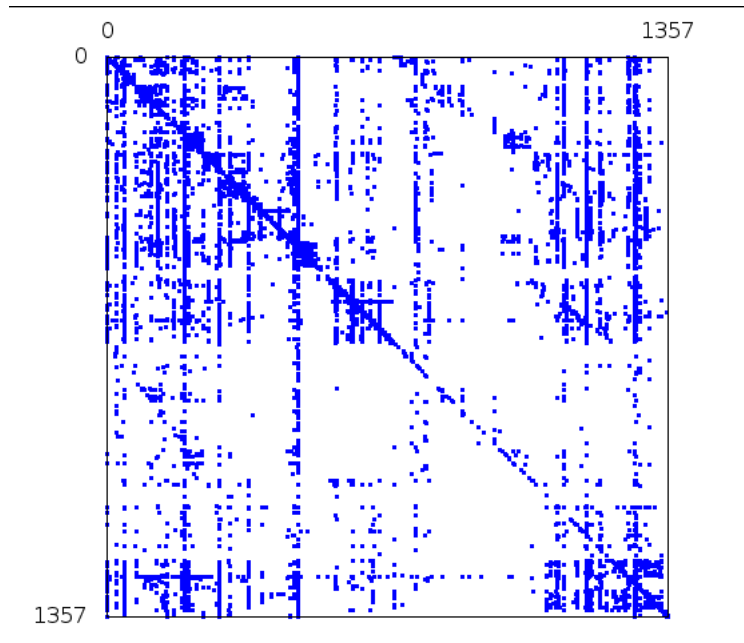


Figura 1: Design Structure Matrix do Linux 2.0 gerado pelo Analizo

provendo:

- Interface Web para visualização de DSM
- Implementação de algoritmos de clusterização de DSM
- Interação com zoom e reorganização da DSM em clusters

Deverá ser possível através da interface Web selecionar uma entre várias sugestões de organização da DSM em cluster calculada automaticamente pela ferramenta. A organização padrão será cluster com base na organização de diretórios dada ao projeto pelos seus desenvolvedores.

Esta ferramenta irá prover ainda a infraestrutura para que usuários cadastrem endereços de repositórios de código-fonte que serão então automaticamente copiados para serem então analisados e disponibilizados para visualização e interação com a DSM.

Outras referências sobre DSMs: [4, 5, 6, 7, 8, 9, 12, 13].

Esta ferramenta irá utilizar como backend o projeto Analizo, uma suíte de ferramentas de análise e visualização de código fonte que suporta C, C++ e Java. Esse projeto possui suporte ao cálculo de mais de 20 métricas de software, além de poder analisar e armazenar tanto métricas quanto informações estruturais do código-fonte de cada revisão armazenada em um repositório de controle de versão.

O software Analizo já possui suporte a algumas visualizações, como a própria DSM, mas nesse trabalho pretendemos implantar uma nova visualização que permita a interação com o usuário, evoluindo assim o software.

A interface para interação com o usuário será desenvolvida utilizando o framework Web NodeJS, uma plataforma para desenvolvimento em Javascript no lado do servidor que permite respostas rápidas às interações. Após o usuário inserir o endereço de um repositório de controle de versão público, o frontend irá se comunicar com o backend, que irá analisar o código fonte para extração de métricas e obter os dados que serão exibidos ao usuário.

Referências

- [1] 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.
- [2] S. D. E. D. E. W. R. P. S. D. A. Gebala, “A model-based method for organizing tasks in product development,” p. 21, 1994.
- [3] N. S. F. Waldman, “Dependency models to manage software architecture,” p. 5, 2005.
- [4] J. L. Rogers, “A knowledge-based tool for multilevel decomposition of a complex design problem,” p. 22, 1989.
- [5] M. J. LaMantia, Y. Cai, A. MacCormack, and J. Rusnak, “Analyzing the evolution of large-scale software systems using design structure matrices and design rule theory: Two exploratory cases,” in *Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*, p. 83, 2008.
- [6] A. de Almeida Souza Neto, “Antares dsm: Visualização e otimização de dependências em design structures matrices,” p. 50, 2008.
- [7] T. R. Browning, “Applying the design structure matrix to system decomposition and integration problems: A review and new directions,” vol. 15, 2001.
- [8] R. A. B. J. F. D. G. J. de Souza Santos; Adauto Trigueiro de Almeida Filho; José Martins da Nóbrega Filho; Jorge César Abrantes de Figueiredo; Dalton Dario Serey Guerrero, “Design suite: Towards an open scientific investigation environment for software architecture recovery,” p. 8, 2009.
- [9] A. K. J. Wang, “Efficient organizing of design activities,” p. 17, 1997.
- [10] D. P. J. C. C. S. D. C. Eckert, “Predicting change propagation in complex design,” p. 10, 2001.

- [11] “Predicting requirement change propagation using higher order design structure matrices: An industry case study,” p. 42, 2011.
- [12] A. F. H. K. Wimmer, “Reachability matrices and cyclic matrices,” *Electronic Journal of Linear Algebra*, p. 8, 2010.
- [13] B. H. Kevin J. Sullivan; William G. Griswold; Yuanfang Cai, “The structure and value of modularity in software design,” p. 10, 2001.
- [14] M. C. S. D. E. D. E. Whitney, “Using the design structure matrix to estimate product development time,” p. 10, 1998.