Instructions for Authors of WDCOPIN

Rodrigo Souza[1]

[1]Departamento de Sistemas e Computação – Universidade Federal de Campina Grande (UFCG)
Caixa Postal 10.106 – CEP 58.109-970 – Campina Grande – PB – Brazil

rodrigo@dsc.ufcg.edu.br

Abstract.

Resumo.

1. Introdução

A divisão conceitual de um sistema de software em módulos é uma informação valiosa durante o seu desenvolvimento, sobretudo para a divisão de tarefas entre desenvolvedores [cite]. Essa informação muitas vezes não é documentada adequadamente e, à medida que os desenvolvedores são substituídos, pode se perder.

Algoritmos de modularização de software procuram identificar a organização de um sistema de software a partir de sua implementação e, por isso, auxiliam desenvolvedores a recuperar a organização modular de um sistema sobre o qual se tem pouco conhecimento. Para realizar a divisão em módulos, alguns algoritmos usam heurísticas baseadas em identificadores, ou então informações sobre o histórico de desenvolvimento (CVS). Muitos algoritmos, no entanto, consideram apenas o conjunto de dependências entre os componentes da implementação do sistema.

Sistemas de software, no nível de implementação, são formados por vários componentes que interagem entre si. Programas orientados a objetos, por exemplo, são compostos por classes que interagem com outras classes através de mecanismos como herança e agregação. As interações estabelecem relações de dependência entre os componentes, de forma que cada componente só funciona corretamente na presença dos componentes dos quais ele depende. Essas dependências formam uma rede de software, na qual os vértices representam as classes e as arestas representam dependências.

A rede de dependências entre componentes de um sistema pode ser extraída automaticamente através de ferramentas de análise estática de código-fonte ou código objeto. A rede extraída deve ser considerada apenas uma aproximação da rede de dependências real, pois nenhuma técnica automática é capaz de determinar corretamente, em todos os casos, se existe uma dependência entre dois componentes [cite].

Estudos recentes analisaram redes de software sob a luz da teoria das redes complexas e encontraram propriedades topológicas marcantes que também estão presentes em redes de interações entre proteínas, redes sociais e muitas outras [cite]. Descobriu-se que nessas redes a distribuição dos graus dos vértices é bem aproximada por uma lei de potência $N(k)\ k^{-\gamma}$, isto é, o número de vértices ligados a exatamente k arestas é proporcional a uma potência de k com expoente negativo e constante. Redes caracterizadas por essa distribuição de graus específica foram chamadas de redes livres de escala.

A partir dessa observação foram desenvolvidos diversos modelos estocásticos que produzem redes livres de escala.

2. Identificação do Problema

Espera-se que os algoritmos de modularização de software encontrem organizações modulares semelhantes àquelas que seriam encontradas por especialistas nos sistemas analisados. Uma forma de testar os algoritmos consiste, pois, em aplicá-los ao código-fonte de um sistema cuja organização modular esteja documentada e então comparar a organização encontrada com a organização documentada. Infelizmente é difícil encontrar sistemas documentados nesse nível de detalhe e por essa razão a avaliação empírica dos algoritmos ainda é incipiente.

A deficiência na avaliação dos algoritmos de modularização de software é uma barreira para a adoção dos algoritmos na indústria e um obstáculo para o avanço das pesquisas.

3. Objetivos

Para suprir a escassez de sistemas cuja organização modular é documentada, propomos o uso de modelos estocásticos capazes de sintetizar redes de software organizadas em módulos para servirem como conjunto de teste. Para que essa abordagem seja bem-sucedida, no entanto, é preciso que as redes sintéticas sejam realistas, isto é, semelhantes a redes extraídas de sistemas de software reais.

Assim, o objetivo desta pesquisa é analisar modelos estocásticos que possam ser usados para avaliar algoritmos de modularização de software. Esse objetivo pode ser decomposto nos objetivos específicos apresentados a seguir.

Descobrir modelos estocásticos de redes livres de escala organizadas em módulos. A maioria dos modelos disponíveis na literatura ignora a organização modular.

Desenvolver um método para avaliar o realismo de uma rede. Esse método deve encontrar um grau de realismo alto para redes de software reais e um grau de realismo baixo para redes de outros domínios.

Avaliar o realismo dos modelos estocásticos. Um modelo é realista se ele é capaz de produzir redes realistas.

Avaliar algoritmos de modularização através de sua aplicação a um grande número de redes sintéticas. Os resultados poderão ser comparados com resultados experimentais encontrados na literatura.

4. Atividades Planejadas e Realizadas

Foram encontrados na literatura dois modelos que produzem redes organizadas em módulos [cite]. Um terceiro modelo de redes modulares foi desenvolvido a partir da adaptação de um modelo de redes não-modulares [cite].

Para apoiar a avaliação de realismo dos modelos, coletamos um corpo de 65 sistemas de software escritos na linguagem Java. A rede de dependências entre componentes de cada sistema foi extraída através da ferramenta Dependency Finder [footnote]. Coletamos ainda cerca de cem redes de diversos domínios, sobretudo redes sociais e redes biológicas. De cada rede (do domínio de software ou de outros domínios) foram extraídas diversas métricas, tais como número de vértices, número de arestas, número médio de arestas por vértice, o expoente γ da distribuição de graus, entre outras.

Foi encontrada na literatura uma métrica de distância entre duas redes [cite]. Essa métrica foi utilizada em um experimento, descrito em detalhes em um artigo não publicado [footnote], que procurou avaliar o realismo dos modelos. Essa métrica de distância foi, no entanto, descartada, uma vez que indicou que uma rede biológica está mais próxima de redes de software do que as redes de software estão próximas entre si.

No momento estamos procurando outro método para avaliar o realismo de redes e, consequentemente, de modelos.

Com isso pronto, queremos publicar. Afinal, o setup experimental está pronto.

A seguir pretendemos avaliar algoritmos de modularização de software.

E então escrever a dissertação.

Por fim, publicar

5. Resultados Obtidos

Desenvolvemos o modelo BCR+, que produz redes orientadas modulares que satisfazem a uma arquitetura modular fornecida como parâmetro. Ele foi baseado em um modelo de redes livres de escala orientadas sem módulos [cite]. Encontramos ainda dois outros modelos de redes orientadas modulares, o LR e o CGW, e um modelo de rede modular não-orientada, o LFR.

Desenvolvemos um procedimento experimental para responder à seguinte pergunta: os modelos são capazes de produzir redes semelhantes a redes extraídas de sistemas reais? Consideramos que um modelo é bem sucedido se ele é capaz de sintetizar uma rede cuja distância para uma rede real tomada como referência não seja maior do que a distância entre redes reais com o mesmo número de vértices da rede de referência.

Nesse experimento tomamos como referência a rede do jogo VilloNanny 2.2.4, com 1658 vértices. Avaliamos os modelos BCR+ e LR.

avaliar o realismo das redes de software sintéticas, baseado na distância de Andrade. Esse procedimento foi executado para avaliar o realismo. O procedimento foi descrito em um artigo ainda não publicado.

Os dados mostram que o modelo BCR+ foi o que mais se aproximou da rede de referência; uma das redes geradas ficou mais próxima à rede de referência do que a rede do sistema JFreeChart. O modelo LFR produziu redes mais próximas entre si, como revela o pequeno desvio-padrão, porém mais distantes da rede de referência.

Estudo qualitativo e quantitativo sobre a verossimilhança dos modelos. Mostrar resultados experimentais: - Medição de frequência de motifs. - Mostrar tabela de distâncias. - Mostrar tabela das métricas (distribuição de graus etc). - Mostrar gráfico da distribuição de motifs e mostrar que há uma diferença visual.