MinerAll: Uma ferramenta para extração e mineração de dados de repositórios de software livre

José Teodoro da Silva[1], Igor S. Wiese[1], Igor Steinmacher[1], Marco Aurélio Gerosa[2]

[1] Coordenação de Informática – Universidade Tecnológica Federal do Paraná (UTFPR)
BR 369 - km 0,5 87301-006 - Caixa Postal: 271 - Campo Mourão - PR

[2] Departamento de Ciência da Computação – Universidade de São Paulo (USP)
Rua do Matão, 1010 - CEP 05508-090 - São Paulo - SP

joseteodoro1105@gmail.com, {igor, igorfs}@utfpr.edu.br,
gerosa@ime.usp.br

Abstract. *Free open source software (FLOSS) projects make their source code available in Version Control Systems (SCMs), providing large amount of data for scientific research. This paper presents the tool MinerAll, which aims to mine data from Open Source projects to investigate the collaboration among developers and the degree of dependency among artifacts. This tool presents as advantage of being flexible, allowing the design and coupling of modules to mine different types of SCMs, and also to allow storage in different media (databases, file systems, etc.). Some preliminary results are also presented.*

Resumo. *Os projetos de software livre de código aberto (FLOSS) disponibilizam seus códigos fontes em Sistemas de Controle de Versões (SCVs) publicamente, fornecendo grande volume de dados para pesquisa científica. Este trabalho apresenta a ferramenta MinerAll, que visa minerar dados destes projetos para investigar a colaboração entre desenvolvedores, o grau de dependência entre artefatos e prever bugs e falhas. Esta ferramenta traz como diferencial a flexibilidade, permitindo a concepção e o acoplamento de módulos para mineração em diferentes tipos de SCV, e de armazenamento em diferentes meios (Bancos de dados, sistemas de arquivos, etc.). Alguns resultados preliminares obtidos também são apresentados.*

1. Introdução

Uma etapa importante de um trabalho científico é o levantamento de dados. As pesquisas na engenharia de software sofreram com a falta de informações reais sobre o processo de desenvolvimento de software. A falta de repositórios de dados dificultava o desenvolvimento de pesquisas na área [Hassan e Xie, 2010]. Entretanto, esse cenário começou a mudar com o crescimento da utilização e do desenvolvimento de software livre de código aberto (*Free/Libre Open Source Software – FLOSS*). Esses projetos disponibilizam seus códigos fontes para o público fornecendo grande volume de dados para a pesquisa científica. Desta forma, é possível obter dados que incluem histórico de alteração de códigos fontes obtidos de Sistemas de Controle de Versões (SCVs), listas de discussão, fóruns e histórico de mecanismos de controle de defeitos (*bugs*).

De acordo com Balieiro et al. (2007) os projetos FLOSS têm sido alvo de estudos nas mais diversas áreas de conhecimento, como, por exemplo: o processo de software

adotado pelas comunidades de software livre [Jensen e Scacchi, 2007]; as práticas de qualidade adotadas [Haloran e Scherlis, 2002]; e os aspectos sociais deste tipo de prática [de Souza et al., 2005].

No que se refere aos aspectos sociais do desenvolvimento de software, atenção especial tem sido dada a técnicas baseadas na análise de redes sociais (*social network analysis*) [Wasserman e Faust, 1997], [de Souza et al., 2005]. Este tipo de rede enfoca os relacionamentos que surgem entre os desenvolvedores de uma comunidade, ao invés de focar em atributos individuais dos desenvolvedores [Balieiro, et al., 2007]. Os estudos relacionados a redes sociais são focados em mineração e extração de informações de repositórios de artefatos de software, para descoberta de processos, relacionamentos entre artefatos e desenvolvedores, e identificação de pontos de coordenação e cooperação entre os desenvolvedores.

Segundo Trainer et al. (2005), pesquisadores e técnicos reconhecem há um longo tempo que problemas de comunicação e esforços de coordenação constituem um grande problema no desenvolvimento de software. Isto se deve, basicamente, à forte interdependência entre os módulos que compõem o software, às atividades do processo de desenvolvimento e às pessoas participantes desse processo.

Neste contexto, este artigo tem como objetivo apresentar a ferramenta MinerAll, para mineração de dados de repositórios de códigos fontes (SCVs), focando a extração de informações de redes de colaboração destes projetos. As redes extraídas pela ferramenta são: (i) Redes sociais: rede de interações entre indivíduos durante o desenvolvimento; (ii) Redes técnicas: rede de relacionamentos existentes entre artefatos; (iii) Redes sóciotécnicas: rede de interações sociais geradas pelo trabalho em equipe sobre os artefatos de software [Ducheneaut, 2005]. A ferramenta apresenta uma arquitetura totalmente orientada a padrões de projeto e interfaces possibilitando sua extensão e flexibilidade. Esta flexibilidade permite que novos repositórios e novos mecanismos de armazenamento sejam facilmente criados e acoplados à ferramenta.

2. Trabalhos Relacionados

A mineração e extração de dados de projetos FLOSS tem atraído muita atenção nos últimos anos. Nesta seção são apresentados alguns trabalhos da área, em específico, aqueles que se relacionam à extração de dados com foco em redes de colaboração.

A ferramenta OSSNetwork [Balieiro, et al., 2007] é uma ferramenta que minera dados de repositórios de software a fim de prover visualização para análise de redes sociotécnicas. Para tanto, os dados são extraídos para uma base dados própria, e utilizados pela própria ferramenta. Ariadne [Trainer, et al., 2005] é outra ferramenta com finalidade similar ao OSSNetwork, entretanto a ferramenta oferece diversos tipos de visualização de redes de dependência (técnicas, sociais e sociotécnicas) que podem ser utilizadas para diferentes finalidades dentro de uma equipe. O FLOSSMole [Howison, et al., 2006] fornece uma arquitetura que possibilita o estudo de projetos de software livre, concentrando-se na análise do repositório SourceForge.net.

Existem outros projetos e ferramentas que mineram dados de repositórios de software livre, tais como Tesseract [Sarma, et al., 2009], StarGate [Ogawa e Kwan-Liu, 2008], e SmallBlue [Ehrlich, et al., 2006]. Entretanto, todas as ferramentas apresentadas são

desenvolvidas para fins específicos, não oferecendo formas de flexibilizar o meio de armazenamento ou o tipo de repositório que se pretende minerar. De maneira geral, as ferramentas existentes permitem que os dados sejam extraídos de um SCV específico, ou de um subconjunto de SCVs. A ferramenta MinerAll, apresentada aqui, é extensível, permitindo a concepção de módulos de mineração de dados para outros SCVs, novos meios de armazenamento dos dados minerados, novos algoritmos de mineração de informações e representações gráficas dos dados.

3. A ferramenta MinerAll

Nesta seção será apresentada a ferramenta MinerAll, uma ferramenta flexível, para mineração de dados dos repositórios de software livre. A versão corrente da ferramenta está disponível em http://minerall.googlecode.com, sob licença GNU GPL v3.

3.1. Arquitetura

A fim de acessar os dados disponibilizados pelos projetos FLOSS, a arquitetura de um minerador foi definida para coletar os dados de histórico de mudança dos artefatos da internet e armazená-los localmente. Esta aplicação pode processar os dados adquiridos salvando-os e/ou gerando gráficos dos resultados. A Figura 1 mostra como o minerador foi desenvolvido. Ele é composto de quatro módulos que serão descritos no decorrer desta seção.

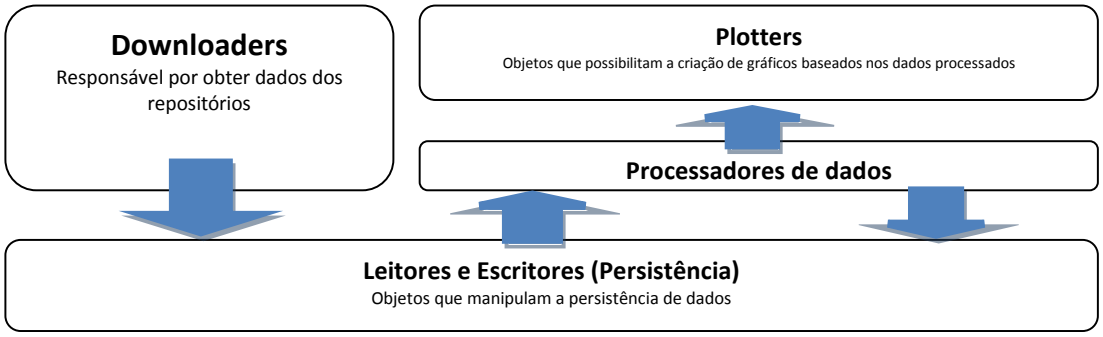

Figura 1. Arquitetura de alto nível da ferramenta MinerAll

Os *Downloaders* são entidades capazes de ligar-se a um repositório e adquirir seu histórico. São recuperadas informações relativas aos metadados dos arquivos (nome, tipo, tamanho, entre outros) e com relação ao *commit* em si (usuário responsável, data, tipo do envio, etc.). Após esta coleta, o módulo de persistência é acionado a fim de armazenar os dados adquiridos.

O módulo de persistência foi idealizado como sendo uma estrutura de Leitores e Escritores extensíveis. Os Escritores são utilizados para persistir os dados adquiridos dos repositórios pelos *Downloaders*. Eles recebem um objeto contendo os dados adquiridos e provém o mecanismo de persistência para estes dados. Já os Leitores são capazes de recuperar os dados armazenados pelos escritores.

É importante ressaltar a possibilidade de criação de novos Escritores/Leitores que manipulem diferentes estruturas de armazenamento de dados, e alternar entre eles em tempo de execução. Atualmente, a arquitetura utiliza tanto o sistema de arquivos como bancos de dados. No entanto, a arquitetura permite a flexibilização adicionando novos

mecanismos de manipulação como *plugins*, uma vez que uma interface bem definida é fornecida. As funções de escrita dessa estrutura são utilizadas pelo minerador para persistir os dados adquiridos dos repositórios. As funções de leitura são mais utilizadas para processamento de dados ou geração de gráficos.

Para a criação dos gráficos, criou-se uma estrutura de Plotters e Processadores de dados que trabalham conjuntamente com os Leitores. Esta estrutura é apresentada na Figura 2. Os processadores de dados adquirem os dados pelos objetos de leitura e trabalham sobre eles. O processamento dos dados gera como saída matrizes numéricas de relacionamento. Um exemplo de matriz gerada é a de usuário *versus* artefatos, que possibilita verificar quais são os artefatos em que os usuários já atuaram, e quais são os usuários já alteraram um determinado artefato.

As matrizes são fáceis de manipular computacionalmente, no entanto, são de difícil interpretação para os humanos. Por este motivo são encaminhadas aos Plotters que, transformam as informações tabulares em visualizações gráficas para um melhor entendimento. Atualmente estão implementadas visualizações em formatos de gráficos de barras, circulares, de frequência e de distribuição.

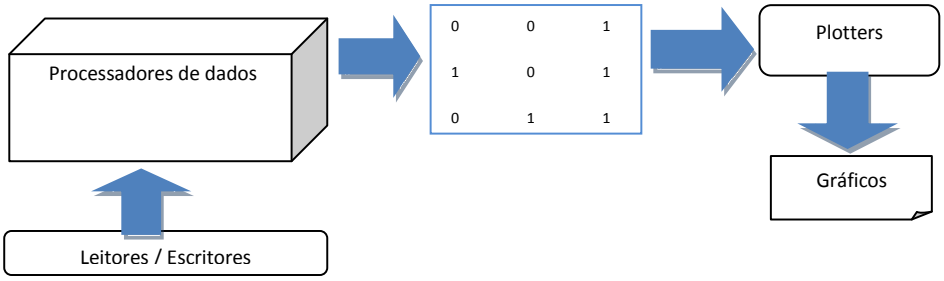

Figura 2. Processo de criação de gráficos a partir dos dados minerados

Apesar dos processadores de dados serem comumente utilizados durante a geração de gráficos, é possível utilizar Escritores para persistir os resultados. Esta funcionalidade é útil em casos onde o processamento de dados para um gráfico seja muito dispendioso. Nestes casos, o processamento pode ser fracionado em partes e serem realizadas sequencialmente. Ao fim de cada parte, os dados devem ser persistidos para garantir a consistência dos resultados. Esta persistência possibilita a continuação de um processamento após uma interrupção inesperada.

3.2. Tipos de dados extraídos

A ferramenta extrai o histórico das modificações realizadas nos repositórios. Até o momento foram levantados dados de projetos que estão armazenados em dois tipos de SCV: Subversion (SVN) e CVS. São extraídos todos os dados relativos às alterações realizadas e enviadas aos repositórios, que poderão ser armazenados de acordo com o Escritor projetado, no formato que se fizer necessário.

Com os dados minerados de SCVs, até o momento, é possível realizar análises de dependência de mudanças [Cataldo, et al., 2006] e dos desenvolvedores que se relacionam por meio dos códigos fontes. Estas relações auxiliam na identificação de padrões de colaboração entre os membros que trabalham nestes artefatos. Com esses

dados, é possível verificar os padrões de colaboração implícitos em SCVs e compará-los com a comunicação efetiva, encontrada nos fóruns e sistemas de controle de *bugs*.

3.3 Resultados Preliminares

Até o momento, foi realizada a extração dos seguintes projetos: Apache Derby, Apache Tomcat, PMD (SourceForge), Apache Jakarta JMeter, DrJava (SourceForge) e PDF Split and Merge (SourceForge). A partir destes projetos foram extraídas cerca de 300.000 alterações em artefatos, em aproximadamente 30.000 *commits*, sendo 6300 destes *commits* relacionados à correção de falhas. A primeira análise sobre os dados minerados dos repositórios apresenta uma possível dependência que existe entre pares de classes através do levantamento de *commits* casados ou *co-changes* (classes que são recorrentemente enviadas ao repositório em conjunto). Baseado nestas dependências foram identificados possíveis relacionamentos entre desenvolvedores envolvidos no projeto. Ainda foi possível identificar indícios de redes sociotécnicas a partir das alterações individuais em artefatos postados no repositório.

Com os dados obtidos foi possível identificar um grupo central de pessoas que trabalham mais ativamente nos projetos. Entretanto, os projetos citados representam um inicio das pesquisas realizadas até o momento. Para alcançar resultados mais concretos ainda se faz necessário efetuar a mineração e análise dos dados de um maior número de projetos em busca de padrões de comparação com as informações já obtidas.

4. Conclusão

Apesar de o projeto estar em seu início, o desenvolvimento atual da ferramenta já permite a mineração e interpretação de informações pertencentes aos repositórios de projetos de software livre. A obtenção destes dados atualmente é realizada mediante a informação da URL do repositório SVN juntamente com usuário e senha válidos para aquele repositório. A partir dos dados adquiridos é possível criar representações tabulares (matrizes de participação) e gráficos com participação, frequência de colaboração entre os membros, bem como a frequência e distribuição das correções de bugs. Os dados extraídos dos repositórios fornecem uma base para iniciar análises sobre as redes sociais e o acoplamento dos artefatos de um projeto. Vinculando o número de alterações individuais dos desenvolvedores foram identificados grupos centrais de desenvolvimento de um dado projeto.

A ferramenta desenvolvida ainda necessita ser experimentada e validada formalmente. Ela é utilizada atualmente para pesquisas relacionadas à busca e reconhecimento de padrões para predição de *bugs* e falhas em *builds*. A precisão destes tipos de informações é de grande valia para gerentes e membros de equipes de desenvolvimento de projetos FLOSS, que podem tomar decisões que evitem a ocorrência de falhas.

A arquitetura está sendo aprimorada a fim de criar mecanismos extensíveis de mineração de fóruns de desenvolvedores, listas de e-mails, *bugtrackers*, e outras listas de discussão. A construção destes mineradores fornecerá dados que ampliarão a riqueza das informações que podem ser extraídas. Um módulo responsável pelo levantamento de métricas e visualizações está em fase de análise e projeto, e contará com algumas representações gráficas, tais como, participação de desenvolvedores, frequência de correções, acoplamento físico e lógico entre módulos.

Uma vez que a arquitetura da ferramenta possibilita extensões, podem ser desenvolvidos e adicionados: novos tipos de repositórios, métricas de software, formas de visualização e interpretadores. Com esta ferramenta, é possível estudar o comportamento de desenvolvedores e dos projetos de software livre. Ela fornece informações importantes para tomada de decisões dos líderes de projetos, como a participação dos membros por exemplo. Atividades futuras preveem a ampliação dos mineradores para analisar dados de repositórios e de listas de discussão simultaneamente. Bem como a construção de um serviço web que permita executar consultas remotamente sobre os dados minerados.

Referências

Balieiro, M.A.; Sousa Junior, S.F.; Pereira, L.P.; de Souza, C.R.B. OSSNetwork: Um Ambiente para Estudo de Comunidades de Software Livre usando Redes Sociais. In: Experimental Software Engineering Latin America Workshop, 2007, p. 33-424.

Cataldo, M.; Wagstrom, P.; Herbsleb, J.; Carley, K. Identification of Coordination Requirements: Implications for the Design of Collaboration and Awareness Tools. In: Conference on Computer Supported Cooperative Work (CSCW'06), 2006.

de Souza, C.R.B.; Froelich, J.; Dourish, P. "Seeking the Source: Software Source Code as a Social and Technical Artifact". In: ACM Conference on Supporting Group Work, 2005, pp. 197-206.

Ducheneaut, N.: "Socialization in an Open Source Software Community: A Socio-Technical Analysis". Springer, 2005.

Ehrlich, K., Lin, C.-Y., Griffiths-Fisher, V.: "Searching for experts in the enterprise: combining text and social network analysis", In: ACM Conference on Supporting Group Work , 2007, pp. 117-126.

Halloran, T.; Scherlis, W. "High Quality and Open Source Software Practices". In: Workshop on Open Source Software Engineering, 2002, pp. 19-25.

Hassan, E., Xie, T.: "Software Intelligence: Future of Mining Software Engineering Data". In: Workshop on the Future of Software Engineering Research, 2010.

Howison, J., Conklin, M. and Crowston, K. "FLOSSmole: A Collaborative Repository for FLOSS Research Data and Analyses". Int. J. Inf. Technol. Web. Eng. 2006

Jensen, C.; Scacchi, W. "A Reference Model for Discovering Open Source Software Processes". In: International Conference on Open Source Systems, 2007, p. 11-13.

Ogawa, M., Kwan-Liu, M.: "StarGate: A unified, interactive visualization of software projects", in IEEE Pacific Visualization Symposium (PacificVis), 2008, pp. 191-198.

Sarma, A., Maccherone, L., Wagstrom, P., Herbsleb, J.: "Tesseract: Interactive visual exploration of socio-technical relationships in software development", In International Conference on Software Engineering (ICSE), 2009, pp. 23-33.

Trainer, E.; Quirk, S.; de Souza, C.R.B.; Redmiles, D.F.: "Bridging the Gap between Technical and Social Dependencies with Ariadne". In: Eclipse Technology eXchange (ETX) Workshop, CA, 2005, pp. 26-30.

Wasserman, S.; Faust, K.: "Social Network Analysis: Methods and Applications". Cambridge, UK and New York: Cambridge University Press, 2007.