

6 Conclusão e Trabalhos Futuros

6.1. Trabalhos Relacionados

Evoont (TAPPOLET, 2007) é um conjunto de ontologias para resolver tarefas da disciplina de Análise de Software utilizando ferramentas da Web Semântica. O objetivo é construir um repositório de artefatos de software extensível, flexível e semanticamente anotado que supere as dificuldades encontradas na representação de dados e na integração inter e intra projetos. No entanto, a extração não considera os dados de gerenciadores de dependência e de ferramentas de Integração Contínua. Além disso, as diferenças entre as diversas versões de uma mesma *Abstract Syntax Tree* não são representadas.

SeCold (KEIVANLOO, *et al.*, 2011) é uma plataforma aberta, colaborativa e nos padrões da Linked Data para compartilhar conjunto de dados sobre software. Além de extrair informações de ferramentas gerenciadoras de demandas/defeitos e de sistemas de controle de versão, introduzem uma metodologia para a geração de URIs únicas e reproduzíveis de cada fato extraído. O objetivo é permitir a integração automática entre diferentes conjuntos de dados. Apesar de disponibilizar um grande repositório de informações sobre projeto de software, permitindo a navegação nos dados e consultas, este trabalho não considera informações de ferramentas de integração contínua, as informações detalhadas das diferenças entre duas versões de um elemento de código-fonte e etc., como é feito na abordagem apresentada neste trabalho.

Sourcererdb (OSSHER, *et al.*, 2009) é uma base de dados relacional que armazena informações extraídas de repositórios públicos de projetos de código aberto. Dentre estas informações destacam-se, a *Abstract Syntax Tree* do código fonte e as informações dos arquivos de configuração do Apache Maven. Entretanto, não considera as informações de sistemas de controle de versão e por ser uma base relacional, não é facilmente extensível e seu poder de expressividade é limitado pelos repositórios de software que considera na abordagem.

SEC (Software Evolution Concepts) (HYLAND-WOOD, CARRINGTON, KAPLAN, 2006) é uma ontologia para documentar requisitos funcionais e não funcionais, métricas, resultado da execução de testes e artefatos de linguagem de programação orientada a objetos. Este trabalho tenta acoplar os metadados dos projetos com seus componentes de implementação visando facilitar o entendimento e a manutenção dos softwares. Entretanto, informações sobre versionamento e demandas/defeitos não são contempladas nesta abordagem, conseqüentemente também não considera o relacionamento da estrutura do código fonte com suas revisões.

6.2. Contribuições

Neste trabalho, nós apresentamos uma nova abordagem de mineração de repositórios de software para endereçar alguns desafios presentes no campo de pesquisa de Mineração de Repositórios de Software. O primeiro desafio diz respeito a como criar uma plataforma que seja extensível para novos tipos de repositórios/informações, algo inerente a este campo de pesquisa. O segundo desafio foi desenvolver esta plataforma de modo que ela suporte o grande volume de informações geradas pelas ferramentas utilizadas no processo de desenvolvimento de software. O terceiro e último desafio está relacionado às limitações existentes na expressividade de representação das informações associadas ao código-fonte de um dado projeto, como o suporte a diferentes tipos de linguagens de programação, a representação das diversas versões de um elemento de código-fonte, bem como de seus relacionamentos e dependências. Um exemplo é a relação intrínseca entre as várias revisões no controle de versão de um arquivo e as mudanças na AST do código-fonte contido nela.

Como forma de validar nossa abordagem, nós investigamos especificamente a mineração de informações do código-fonte de um projeto: sua estrutura, versões, testes e dependências. Esta escolha foi baseada no volume de informações destes tipos de dados e no suporte dado à gerentes e desenvolvedores na resposta de perguntas envolvendo estes tipos de conjunto de dados. Exemplos de representação e da execução destas questões foram demonstrados através da criação de consultas SPARQL que utilizam os vocabulários e ontologias

propostos neste trabalho. Mais detalhadamente, este trabalho ofereceu as seguintes contribuições:

1. Uma representação mais precisa e semanticamente significativa dos artefatos de software, incluindo suas interdependências, os tipos de mudança que podem sofrer, testes executados e manutenções em reposta a defeitos/demandas identificados;
2. A integração das dependências dos projetos durante o processo de extração, permitindo um entendimento mais apurado das características do código-fonte examinado;
3. A integração de informações de ferramentas de integração contínua o que permite a análise dos testes e construções executados, incluindo os defeitos e demandas associados;
4. Uma arquitetura que potencializa o paralelismo da mineração de repositórios de software com grande volume de dados.

6.3. Trabalhos Futuros

Deixamos como trabalho futuro a criação de mecanismos para analisar os tipos das licenças obtidas dos projetos como forma de determinar incompatibilidades entre as licenças das várias bibliotecas utilizadas em um projeto. Além disso, a inclusão do suporte para outros tipos de sistemas de controle de versão (ex.: GIT, Mercurial), linguagens de programação (ex.: Python, Ruby, Scala), gerenciadores de dependência (ex.: Gradle, SBT) e ferramentas de integração contínua (ex.: Bamboo, CruiseControl) são possíveis trabalhos futuros. A integração da plataforma com as principais IDEs também será revista futuramente, pois possibilitará a coleta de informações que normalmente só são geradas neste contexto, como a interação do usuário com um determinado trecho de código. Além disso, esta integração poderia auxiliar os usuários nas tarefas apresentadas na seção de Validação, mas sem a necessidade da utilização ou criação de consultas SPARQL. Um exemplo poderia ser o realce dos trechos de código que precisariam ser modificados para atualizar a versão de uma biblioteca utilizada como dependência.

Nós também pretendemos incluir informações relacionadas ao processamento de linguagem natural, como o texto presente em uma issue, os e-

mails das listas de discussão do projeto e os dados de sites como StackOverflow⁹⁰, onde desenvolvedores podem postar ou responder dúvidas sobre um determinado assunto e ganham pontos de reputação por isso. Acreditamos que estão informações não estruturadas podem auxiliar gerentes e desenvolvedores na condução do projeto.

⁹⁰ <http://stackoverflow.com/>