

Bc. Hotovec
Milan

Diplomová práce

Inženýrská informatika
Softwarové inženýrství
2021/2022

Vedoucí práce:

Doc. Ing. Přemysl Brada, MSc., Ph.D.

Analýza závislostí softwarových artefaktů

Abstrakt

Práce je zaměřena na řešení problémů spojených se zaváděním závislostí do aplikací. V rámci práce je řešeno získávání, uložení a následné analýzy grafu závislostí. Cílem práce je navrhnout obecný úložný formát grafu závislostí a navrhnout nástroje pro získávání grafu závislostí a pro jeho následnou analýzu.

Úvod

Využitím existujících komponent při vývoji softwaru vzniká na použitých komponentách závislost. Komponenty mohou mít i své vlastní závislosti a závislosti obecně mohou mít závislosti. Mít závislost znamená, že vyvíjený software bez nich nedokáže fungovat. Je potřeba dokázat analyzovat graf závislostí a předejít tak případným problémům.

Východiska, analytická část

Před realizací návrhu řešení problému byla provedena analýza reprezentace závislostí v různých technologiích, jednalo se hlavně o komponentový systém Nuget pro platformu .NET a systém PyPI pro jazyk Python. Získané znalosti byly použity k lepšímu pochopení jak reprezentovat graf závislostí a jak jej vytvořit.

Na základě studia příslušné literatury a posbíraných podnětů z praxe byla získána množina problémů, které vyžadují analýzu grafu závislostí.

Hlavní aspekty realizace

Významným aspektem realizace je navržení obecného úložného formátu pro uložení grafu závislostí, do modelu je možné uložit také informace o provedených analýzách. Data jsou pomocí výstupu analýz anotována, nad daty je možné provést větší množství analýz a nenarušit tak jejich obsah jinak, než obohacením o příslušné anotace.

Navržená architektura se skládá z jádra, které obsahuje implementaci navrženého obecného úložného formátu a metod pro import a export dat. Další dva bloky, které jsou vázány na jádro jsou nástroje pro získávání grafu závislostí a nástroje pro analyzování.

Možnosti exportu dat mají velký význam pro lepší pochopení získaných dat. Základní export je ve formátu JSON, který je sice čitelný, ale není pro rychlý přehled dokonalý. Další export je možný do textového formátu, kdy se graf vypíše jako stromový výpis s ASCII formátováním, obdobně jako vypadá výstup programu "tree".

Nejvíce užitečný exporter je pro formát dot, jedná se o formát speciálně určený pro grafová data. Tento formát dokonale popisuje graf a je možné jej vizualizovat za pomoci dostupných kompatibilních nástrojů s dot formátem. Mezi vhodné nástroje patří třeba nástroj Graphviz. Ve vizualizaci je možné velmi rychle pochopit strukturu grafu, ve spolupráci s analyzátory, které dodají anotace je zde rovněž rychle vidět, která data jsou anotovaná. Anotovaná data jsou barevně zvýrazněna, modrá barva je informační anotace, žlutá je varování a červená je chyba.

Velkým problémem je získat příslušný graf závislostí, tento problém řeší specializované nástroje pro příslušné komponentové úložiště. Výstupem těchto nástrojů je graf závislostí v obecném úložném datovém formátu.

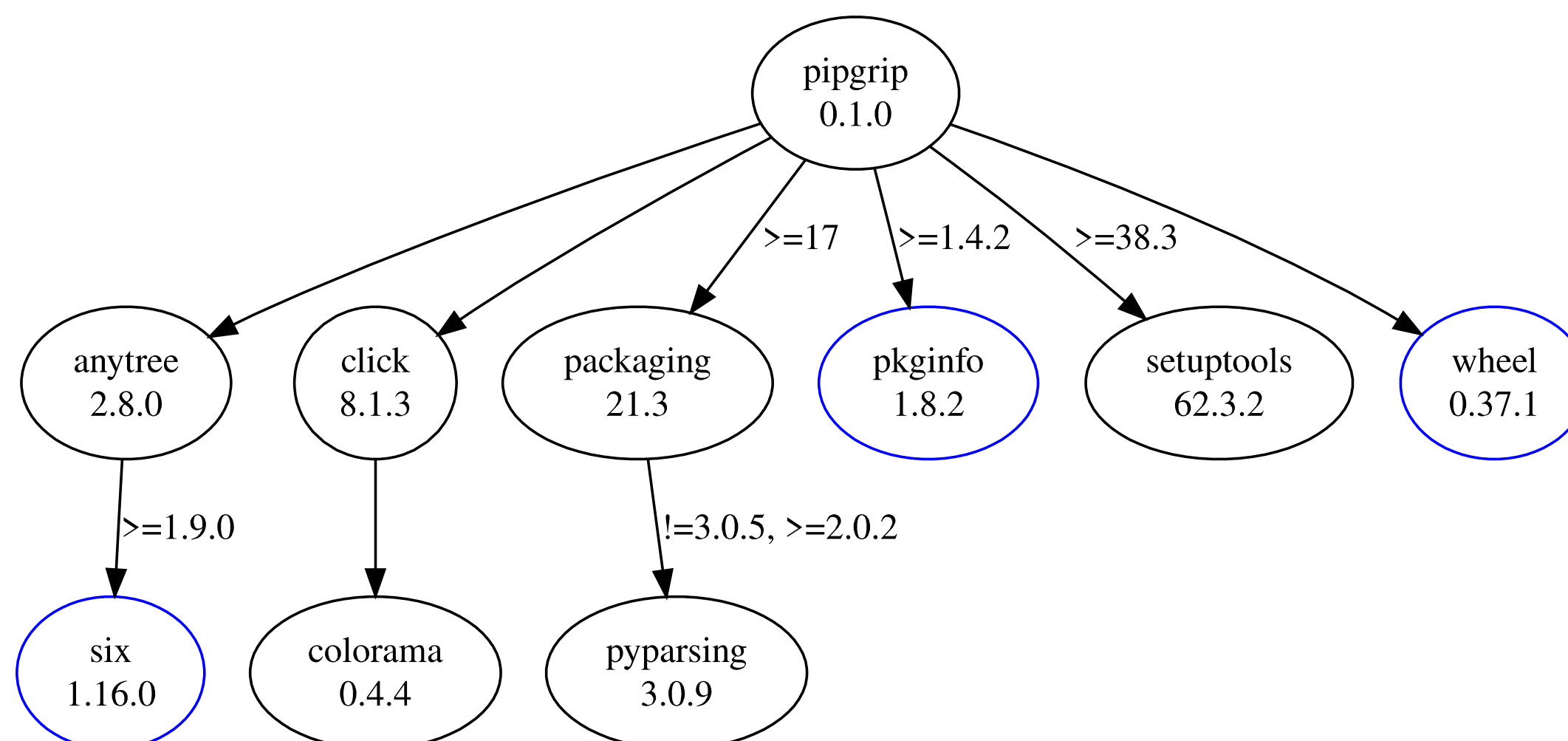
O analýzu a poskytnutí výsledku o specifickém problému spojeném s grafem závislostí se starají specializované nástroje, analyzátory. Jejich výstupem může být anotovaný graf závislostí, případně pouze textová reprezentace výsledků. Výstup je možný podle podstaty provedené analýzy.

Dosažené výsledky

Rozdělením nástrojů dle navržené architektury je velmi snadné vytvořené nástroje dále udržovat a vyvíjet další nástroje. Díky závislosti pouze na jádře nejsou analyzátory ovlivněny implementací ostatních nástrojů a jsou proto obecně využitelné pro libovolné grafy závislostí uložené v obecném úložném datovém formátu popsáném v rámci této práce.

Závěr

Výstupem práce je přístup k řešení problémů spojených s grafem závislostí. Byl navržen obecný úložný datový formát a sada nástrojů, která graf závislostí dokáže vytvořit, ukládat do různých formátů a analyzovat.



Vizualizace závislostí pro jeden balíček