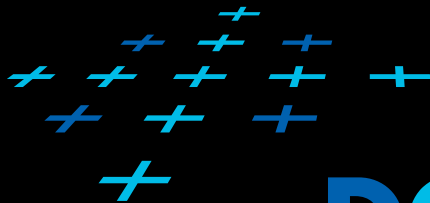


VIZUALIZACE GRAFŮ PRO SOFTWARE ENGINEERING

LADISLAV ČMOLÍK

NTIS – NEW TECHNOLOGIES FOR INFORMATION SOCIETY
ZÁPADOČESKÁ UNIVERSITA V PLZNI

KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE
FAKULTA ELEKTROTECHNICKÁ
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE



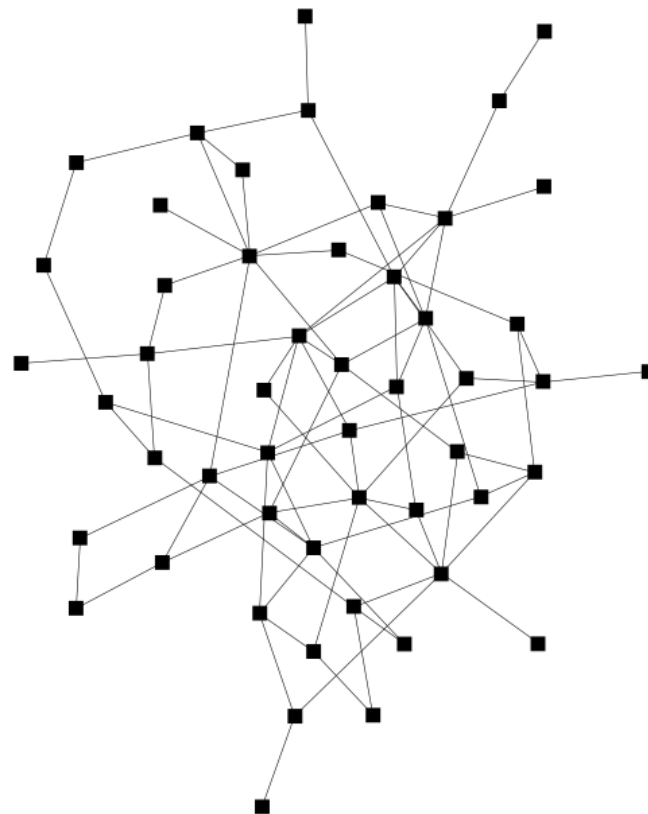
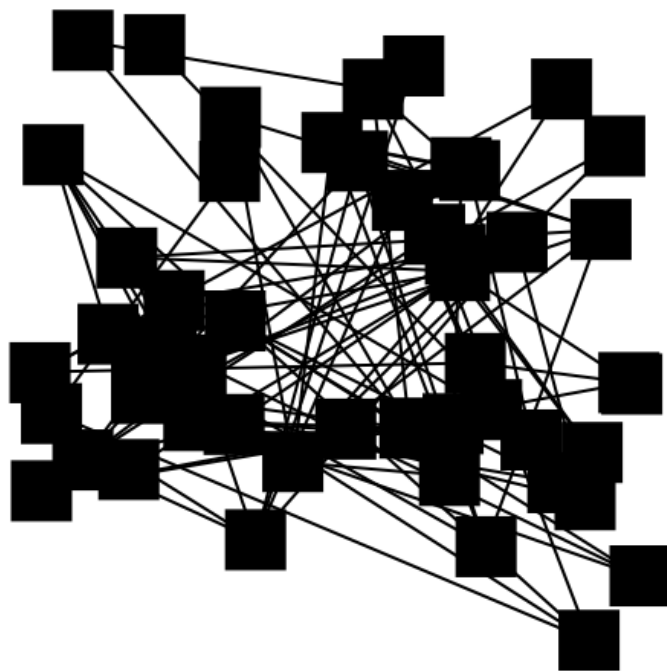
DCGI



- + Komponentové softwarové systémy jsou čím dál složitější
 - + Jejich správa je obtížnější
- + Pomocí reverse engineeringu lze získat model SW systému
 - + Model je stále velmi složitý
 - + Je obtížné se v něm vyznat
- + Je zde potřeba nástrojů, které umožní uživatelům vyznat se v modelu SW systému
 - + Interaktivita
 - + Information hiding
 - + Abstrakce

Layout grafu

- + Layout grafu je rozmístění uzlů grafu
- + Layout grafu může výrazným způsobem ovlivnit vnímání informace obsažené v grafu



+ Špatný layout grafu:

- + Mnoho hran grafu se kříží (lze odstranit pouze pro planární grafy)
- + Uzly spojené hranou jsou daleko od sebe
- + Uzly, které nejsou spojené hranou jsou blízko sebe
- + Hrany vycházející z jednoho uzlu mezi sebou svírají malé úhly
- + Hrany jdoucí stejným směrem jsou blízko sebe

+ Správný (estetický) layout grafu:

- + Hrany grafu se nekříží (nebo se kříží minimálně)
- + Uzly spojené hranou nejsou daleko od sebe
- + Uzly, které nejsou spojené hranou nejsou blízko sebe
- + Hrany vycházející z jednoho uzlu mezi sebou nesvírají malé úhly
- + Hrany jdoucí stejným směrem nejsou blízko sebe
- + Uzly jsou zarovnány v mřížce

- + Existuje celá řada layoutů grafu
 - + Tree layout – vhodné pouze pro stromy
 - + Hierarchický layout – vhodné pro DAG, ale i obecné grafy
 - + Ortogonální layout
 - + **Force-based layout**
 - + **Single-level**
 - + Multi-level

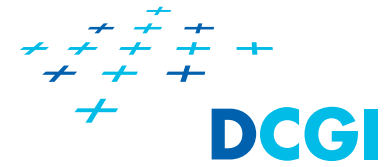
Force-based layout



- + Uzly uvažujeme jako body
- + Síly mezi uzly reprezentujeme
 - + V závislosti na vzdálenosti mezi uzly
 - + Logaritmická, Lineární, Kvadratická
 - + Přitažlivé jako pružiny – Hookův zákon - lineární
 - + Odpuzivé jako sílu mezi dvěma stejně nabitými částicemi - Coulombův zákon - kvadratická
- + Podle toho jaké síly (a jejich velikost) rozmístíme mezi uzly dostaneme různé layouts
 - + Velká flexibilita
 - + Částečná předvídatelnost

- + Uzly, které mezi sebou mají hranu
 - + Přitažlivá síla
 - + Odpudivá síla
- + Uzly, které mezi sebou nemají hranu
 - + Odpudivá síla
- + Závislost síly na vzdálenosti mezi uzly
 - + Logaritmická, Lineární, Kvadratická
 - + Layout se bude lišit dle použitého typu závislosti pro přitažlivé a odpudivé síly
- + Velikost síly
 - + Pokud pro uzel U vynásobíme všechny síly $X > 1$ pak
 - + Uzel U ovlivní uzly, se kterými má hranu stejně jako před tím
 - + Uzly, se kterými nemá U hranu budou dále od U

GPU implementace

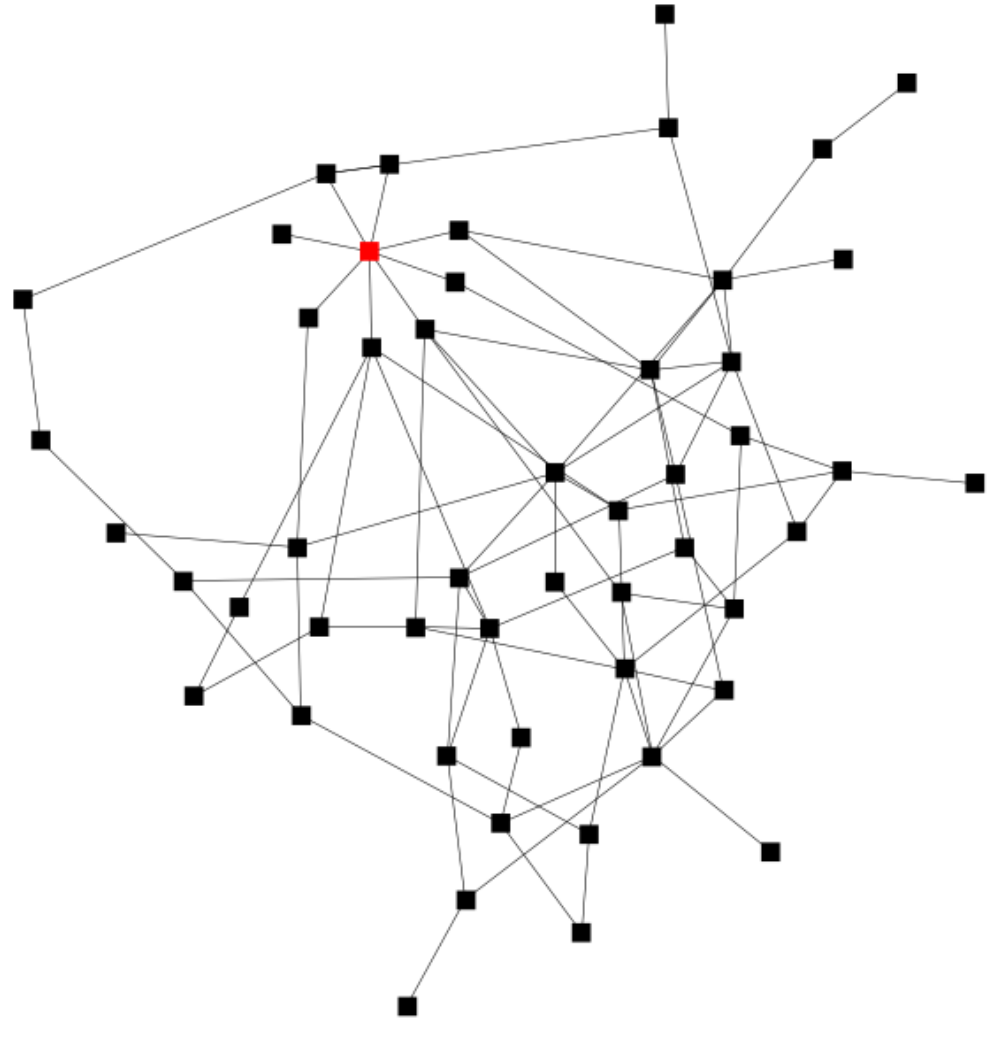
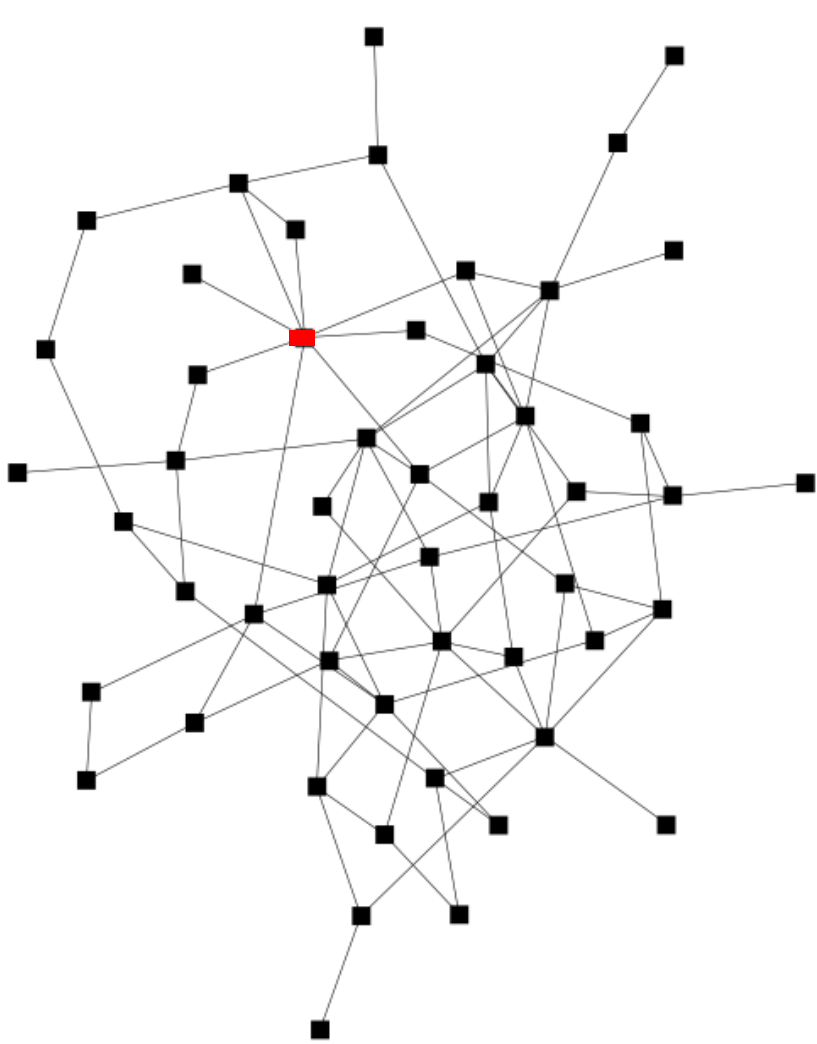


- + Vykreslíme N čar do textury $1 \times N$
 - + N je počet uzlů grafu
- + 1D texturovací souřadnice vrcholů čar jdou od 0 do 1 po $1/N$
- + Při rasterizaci se pro každou čáru vytvoří N fragmentů
- + Pro každé pole textury $1 \times N$ máme N fragmentů
 - + Ty mají různé texturovací souřadnice
- + Pro každý fragment se koukneme do textury (matice sousednosti) dle polohy fragmentu a texturovací souřadnice
 - + Pokud hrana neexistuje fragment zahodíme
 - + Pokud hrana existuje vypočteme sílu
- + Všechny síly sečteme pomocí blendingu

- + Vizualizační technika kdy nezvýrazňujeme jen region zájmu (focus), ale i jeho okolí (context)
- + Na základě *vzdálenosti* od regionu zájmu počítáme tzv. stupeň zájmu (Degree of interest – DOI)
- + Okolí regionu zájmu nezvýrazňujeme stejně, ale na základě DOI
- + *Vzdálenost* může být
 - + Euklidovská
 - + vzdálenost v grafu
 - + Sémantická vzdálenost
 - + atd.

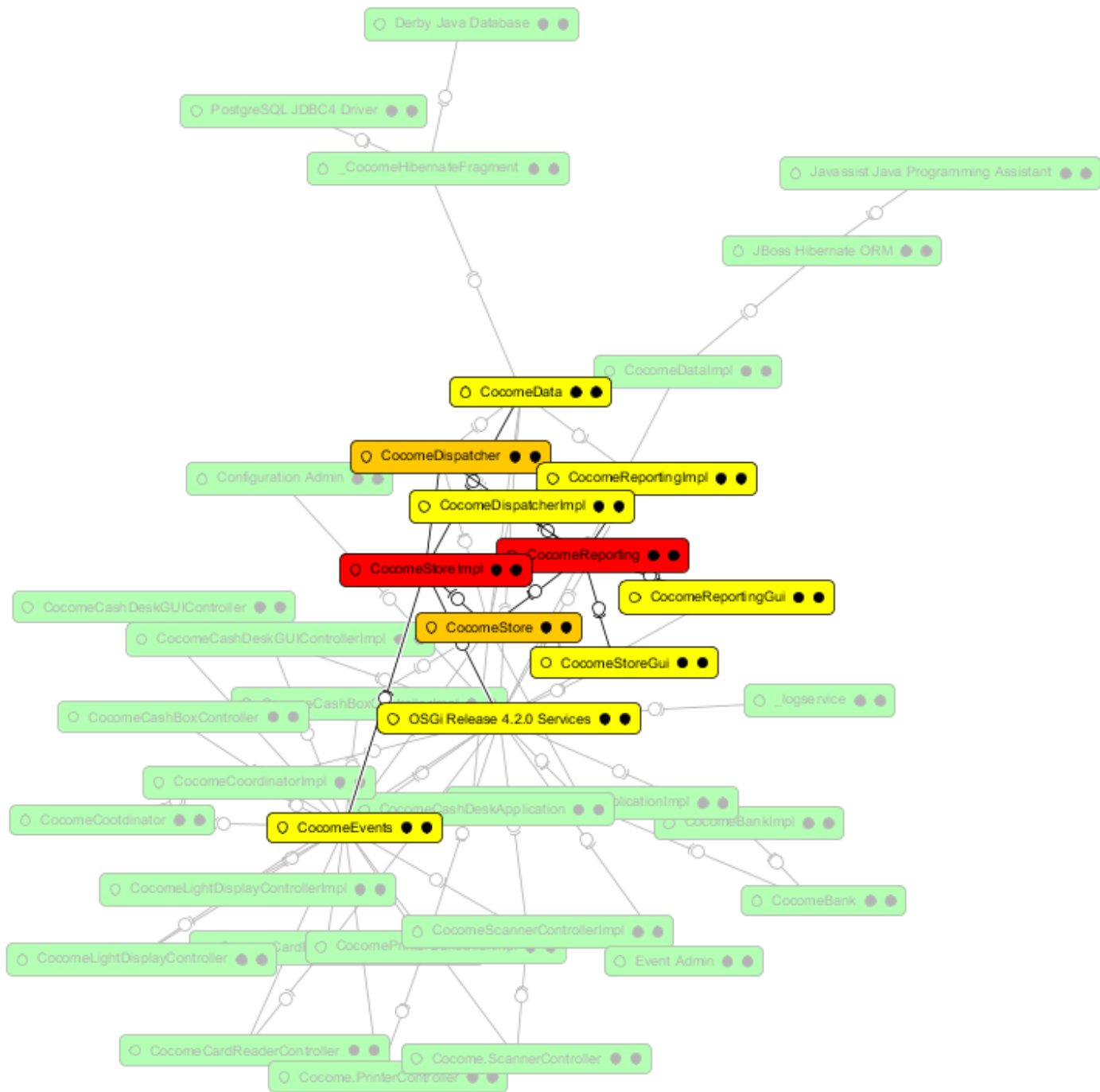
DOI ovlivňuje layout

- + DOI počítáme na základě vzdálenosti v grafu od ROI
- + U komponentových diagramů nás zajímá zejména bezprostřední okolí (sousedé)
 - + Přitažlivé síly mezi ROI a sousedy násobíme DOI
 - + Odpudivé síly mezi ROI a všemi uzly násobíme DOI
- + Velkou nevýhodou je značný pohyb uzlů v grafu
 - + Člověk není schopen sledovat pohyb všech uzlů
 - + Rozbívá se mentální model polohy uzlů



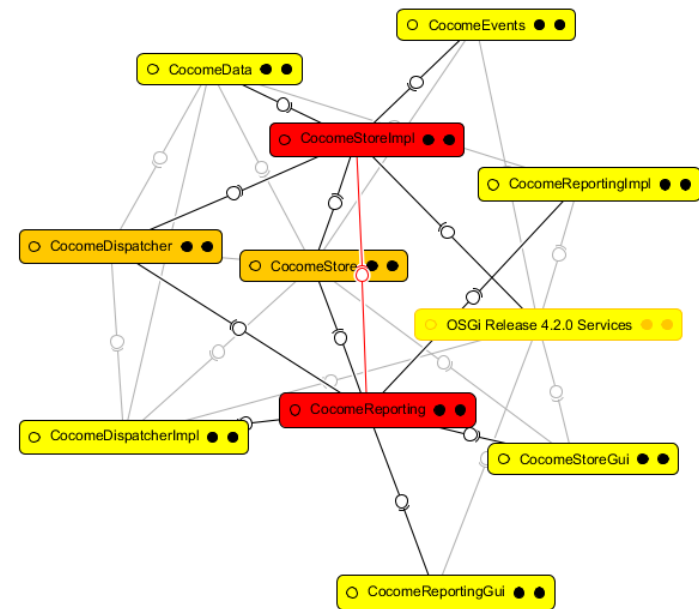
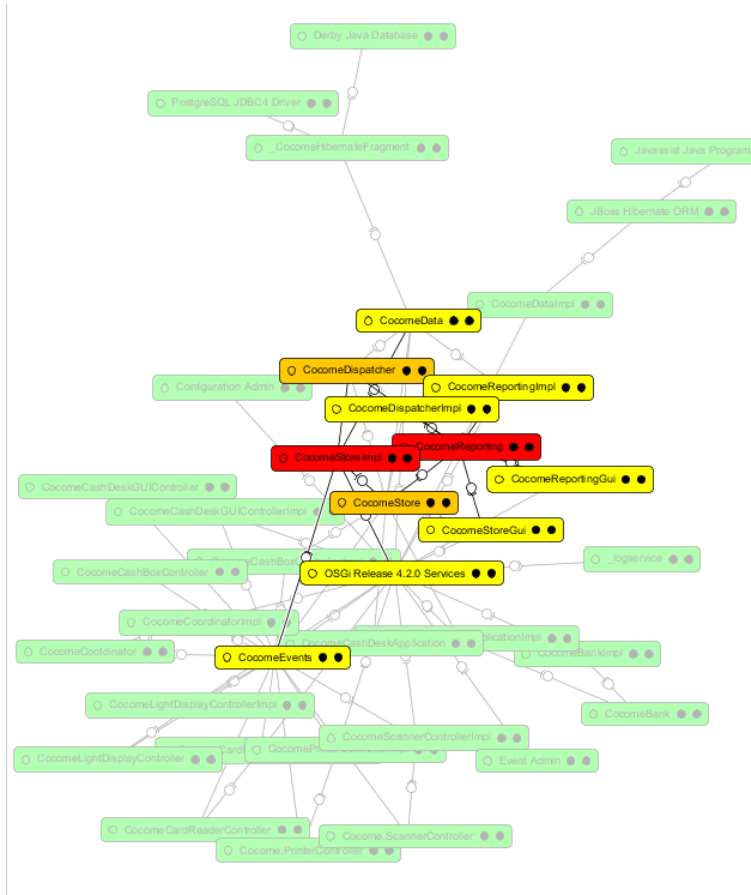
Vizualizace DOI jako barvy uzlu

- + Uvažujeme více regionů zájmu
- + DOI počítáme dle počtu spojení/hran s regiony zájmu (se všemi, alespoň s jedním, s žádným)



Linked views

- + Pro složitější grafy mohou být regiony zájmu a jejich kontext daleko od sebe
- + Zobrazíme graf ještě jednou, ale s jiným layoutem pouze pro regiony zájmu a jejich kontext



- + Metrika – číselné ohodnocení komponenty reprezentující nějakou její vlastnost
 - + Počet spojení s ostatními komponentami
 - + Počet spojení s regiony zájmu
 - + Spolehlivost komponenty
 - + Počet volání jiných komponent při volání komponenty
 - + atd.

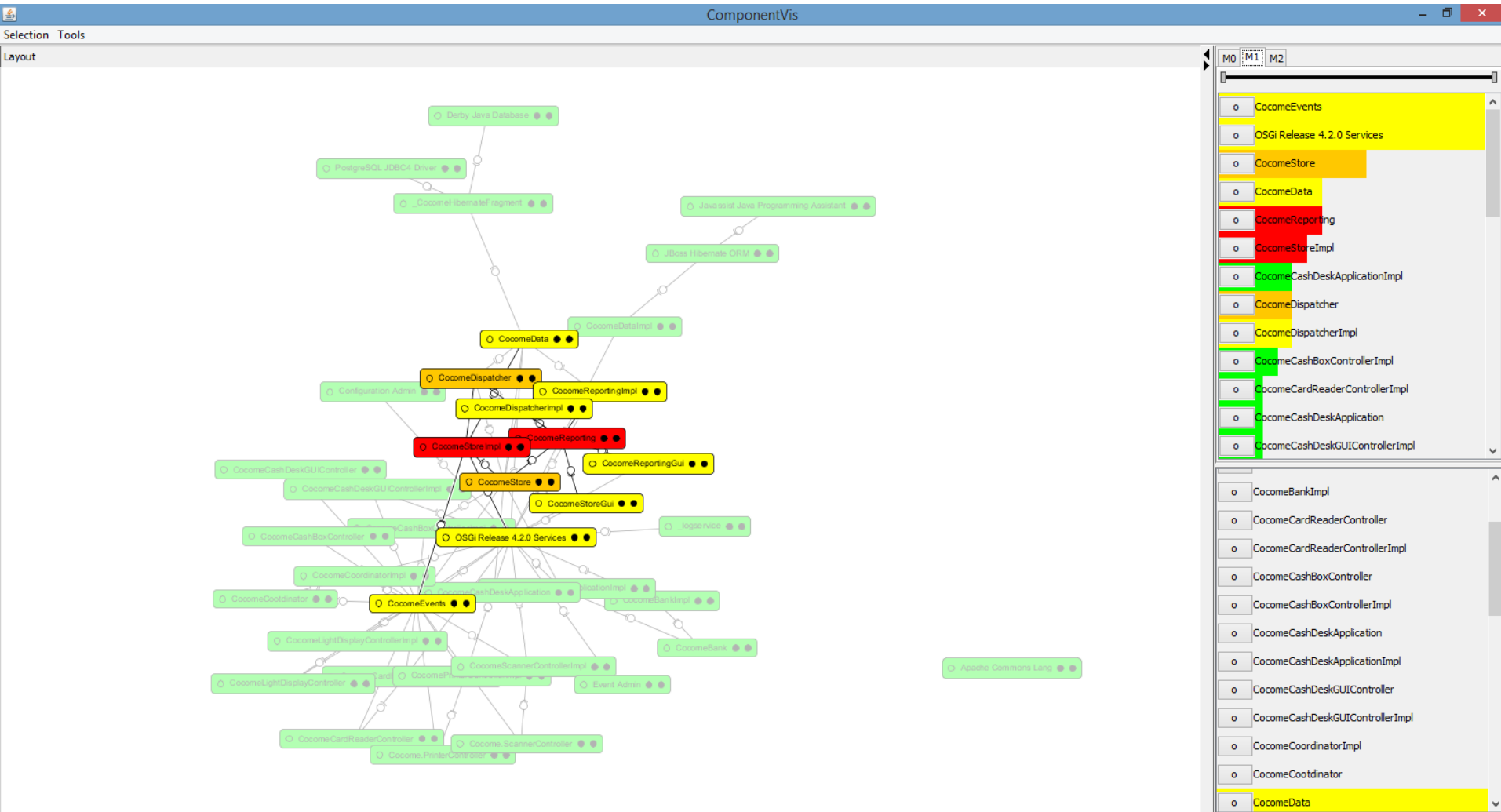
- + Metriky jsou důležité pro analýzu stavu softwarového systému
 - + Komponenta, která je nespolehlivá a zároveň má velký počet spojení s jinými komponentami by měla být opravena co možná nejdříve
 - + Spolehlivá komponenta, která má spojení s mnoha ostatními komponentami není zajímavá, patrně se jedná o rozhraní které všichni implementují

Vizualizace metrik - požadavky

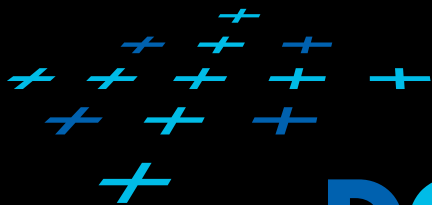


- + Jednotný styl práce s metrikami
- + Možnost filtrovat graf na základě jedné nebo několika metrik
- + Zajímavé jsou převážně mezní případy
 - + Metriky s malou nebo naopak velkou hodnotou
- + Globální pohled
 - + Uživatel by měl mít přehled o rozložení komponent v rámci metriky
 - + Histogram komponent v rámci metriky
- + Předvídatelnost
 - + Uživatel by měl vědět jaké komponenty se vyfiltrují při změně hodnoty filtru
- + Možnost vizualizace metriky v grafu

Vizualizace metrik



Děkuji za pozornost



DCGI

