



SOFTVERSKO REŠENJE ZA PRIKAZ GRAF MODELA  
SOFTWARE SOLUTION FOR GRAPH MODEL PRESENTATION

Momir Marić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PRIMENJENE RAČUNARSKE NAUKE

**Kratak sadržaj** – Dokument uopšteno opisuje problem, strukturu, rešenje i dobijene rezultate softverskog rešenja. Softversko rešenje je u stanju da stvori, rasporedi i prezentuje jedan matematički graf uprošćenog realnog sistema.

**Ključne reči:** Graf, Algoritmi raspoređivanja grafa, Prezentacija grafa

**Abstract** – This paper describes software solution for graph modelling, layout and visualization. The solution provides basic framework to describe and visualize a graph model of a system.

**Keywords:** Graph, Graph layout algorithms, Graph visualization.

1. UVOD

Realni sistemi se modeluju iz dana u dan sa sve većim obimom i kompleksnošću. Mašine su u stanju da prate u stopu novonastale rastuće promene, ali ljudi i ljudska percepcija postaju preopterećeni. Glavni problem aplikacija postaje predstaviti ljudskom korisniku masivan protok količine podataka. Tabele su se pokazale najprikladnije za čuvanje i skladištenje podataka realnog sistema ali ne tako dobre za svakodnevni rad korisnika. Korisniku i ljudskom oku su umesto redova tabela teksta mnogo ugodnije vizuelne reprezentacije tih istih podataka. Vizuelne reprezentacije su bliske oku, prezentuju veliku količinu podataka i najčešće su u obliku slika, grafova, boja, ...

U toku sticanja iskustva u modelovanju i implementaciji aplikacija, uočen je mali broj komponenti za prikaz matematičkih grafova. Jednostavan matematički graf (u nastavku rada se koristi samo graf) je uređen par  $G = (N, E)$ , gde je  $N$  konačan, neprazan skup čvorova, a  $E$  je skup grana koje predstavljaju povezanost čvorova.

Za temu rada odabrani su dizajn i implementacija WPF komponente za stvaranje modela grafa, njegovo raspoređivanje i na kraju prezentaciju dobijenog modela. Komponenta se instancira u XAML kodu gde joj se prosleđuje model grafa kojeg prikazuje. Za ispunjavanje ciljeva komponente modelovana je implementacija grafa koja se dobija učitavanjem XML direktorijuma.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivetić, red. prof.

Raspoređivanje elemenata vrši se uz pomoć spoljašnje biblioteke. Razvijanje sopstvenog algoritma raspoređivanja je veliki obim za svrhe ovog rada i time se neće biti bavljeno. Odabrana je biblioteka MSAGL. Biblioteka podržava Sugiyama, MDS i Incremental algoritme [2]. Nakon raspoređivanja, elementi se prezentuju tako što se iscrtavaju na platnu uz pomoć XAML opisa za svaki odgovarajući element grafa.

2. MODEL GRAFA

Softversko rešenje teži ka integraciji sa većim brojem biblioteka za raspoređivanje elemenata. Za potrebe navedene integracije model grafa je implementiran koristeći interfejs. Izdvojeni su atributi čvora neophodni za raspoređivanje: granični okvir, širina, visina, koordinata tačke i reference na ivice preko kojih je jedan čvor povezan sa drugim čvorovima. Atributi ivice su reference ka prvom i drugom čvoru i opciono naziv ivice.

Granični okvir je poligon koji čvor obuhvata. Softversko rešenje podržava bilo koji poligon, ali se radi jednostavnosti implementacije i integracije sa drugim bibliotekama koristi samo pravougaonik. Za koordinatnu tačku odabrana je gornja leva tačka pravougaonika. Inicijalno svi čvorovi poseduju koordinate nulte tačke i tek nakon algoritma raspoređivanja dobijaju svoje koordinate na platnu.

Struktura grafa se dobija učitavanjem elemenata iz XML fajla. Povezanost se dobija vezama predak-potomak i referentnog elementa. Veze predak-potomak preslikavaju elemente koji su ugnježdjeni u druge elemente. Veza referentnog atributa se dobija tako što se jedan element proglasi za referentni. Svaki element koji u sebi sadrži referentni element biće preslikan na čvor grafa. Na osnovu vrednosti referentnih atributa prvi čvor će biti povezan sa drugim čvorom ukoliko prvi čvor sadrži vrednost bilo kojeg elementa koja odgovara vrednosti referentnog atributa drugog čvora. Za naziv novonastale ivice se uzima naziv elementa prvog čvora.

Prethodno dobijen model grafa se pretvara u model grafa koji odgovara biblioteci koja će izvršiti raspoređivanje elemenata grafa. Preslikani model se raspoređuje prema implementaciji nekog od algoritama raspoređivanja kao što su Sugiyama i MDS.

Dobro raspoređen graf sadrži minimalan broj preklapanja i presecanja svojih elemenata. Ne postoji univerzalan algoritam raspoređivanja koji je u stanju da najbolje rasporedi bilo koji model grafa realnog sistema. Softversko rešenje je moguće upotrebiti kao alat za

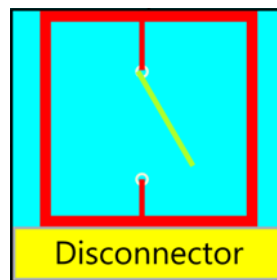
pronalaženje algoritma koji najviše odgovara jednom realnom sistemu. Pored pronalaženja odgovarajućih algoritama, softversko rešenje može da prikaže i rezultate različitih implementacija algoritama raspoređivanja.

Nakon izvršavanja algoritma raspoređivanja, model grafa se ponovo konvertuje nazad u model grafa softverskog rešenja i prikazuje.

### 3. PREZENTACIJA GRAFA

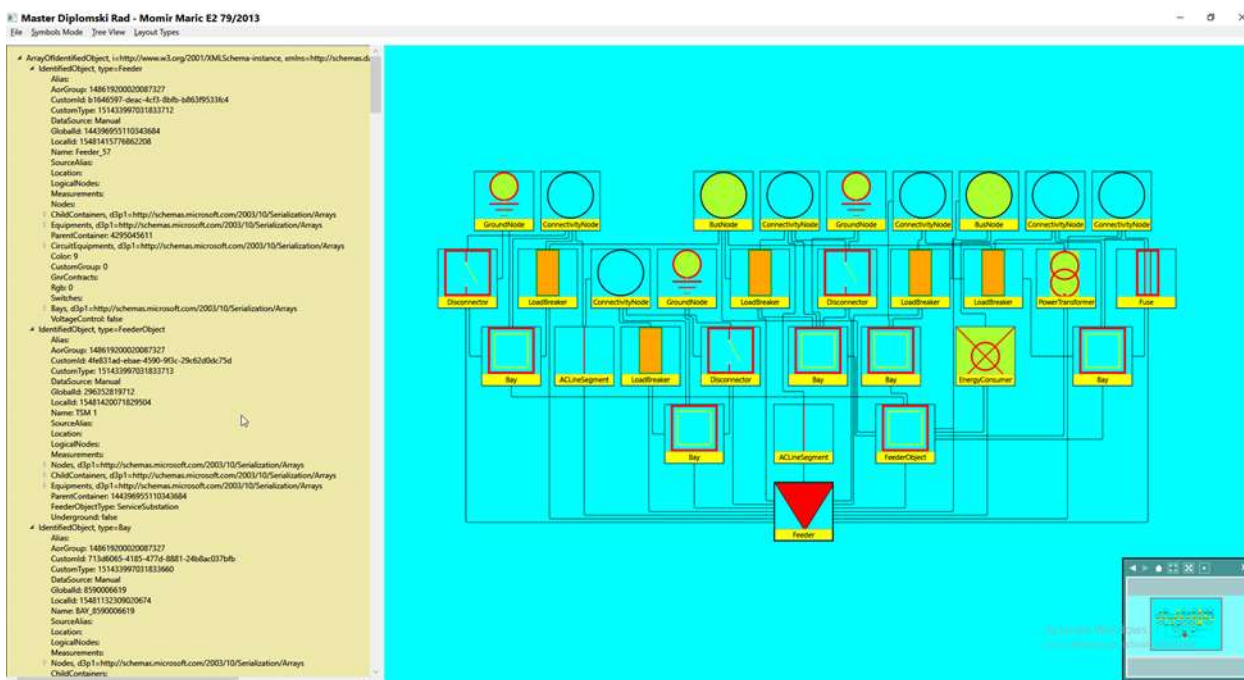
Nakon stvaranja i raspoređivanja, elementi grafa se prikazuju na platnu. Svaki element koji se prikazuje poseduje sopstveni XAML opis.

Opis se sastoji iz skupa geometrijskih oblika iscrtanih na platnu na osnovu širine, visine i koordinatne tačke čvora grafa. Osnovne oblike koje XAML podržava su: elipsa, linija, putanja, poligon, skup linija i pravougaonik [1].



Slika 1. Primer elementa

Na slici 1 prikazan je primer implementacije opisa jednog elementa realnog sistema. Opis se sastoji od ivičnog pravougaonika, centralnog obojenog pravougaonika i dve elipse i tri linije koje opisuju unutrašnje oblike.



Slika 2. Rezultat prikaza grafa realnog sistema

Slika 2 prikazuje konačan rezultat softverskog rešenja. Na levoj strani u obliku drveta je prikazan učitani XML direktorijum koji se prikazuje, a na desnoj strani su prikazani pojedinačni elementi raspoređeni Sugijama algoritmom. Platno je moguće pomerati i zumirati. U donjem desnom uglu nalazi se globalni pregled platna.

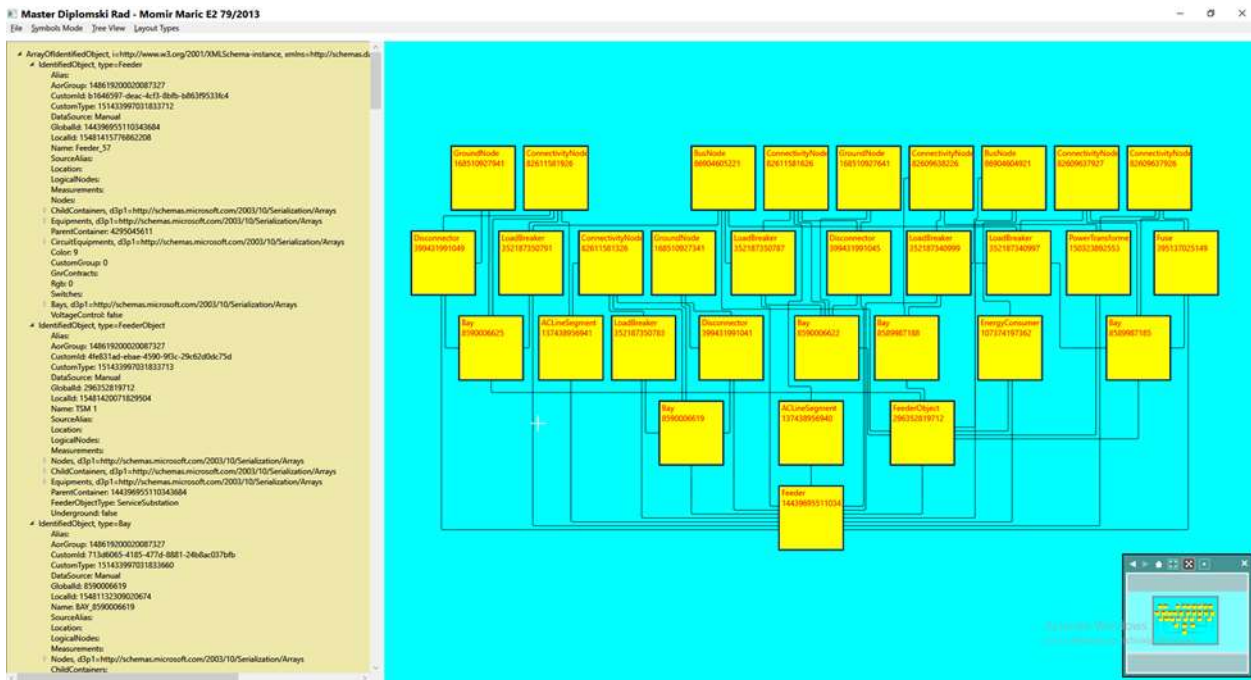
Softversko rešenje je u stanju da podrži prikaz više različitih realnih sistema, ili prikaz istog sistema na različite načine u vidu iscrtavanja različitih simbola elemenata ili prikazivanja različitih povezanosti elemenata odabirom drugačijih referentnih tački. Takva osobina se podržava stvaranjem XML profila. XML profil sadrži skup XAML opisa i informaciju koji XML elementi su referentni.

Na slici 3 prikazan je implementirani tekstualni profil sa istim grafom i opcijama raspoređivanja kao na slici 2. Svi elementi se iscrtavaju univerzalnim simbolom koji se sastoji iz osnovnog graničnog okvira i teksta elementa čiji

sadržaj odgovara podacima XML elementa kojeg prikazuje.

Čvorovi grafa sadrže atribute tipa na osnovog kojeg se određuje koji XAML opis će se koristiti za njegovo iscrtavanje. Ukoliko element nema grafičku reprezentaciju u odabranom XML profilu, on će biti iscrtan tekstualnim profilom.

Softversko rešenje podržava sistem plugin-a. Aplikacija se može proširiti novim algoritmima raspoređivanja i XML profilima jednostavno tako što se implementacija interfejsa smesti na odgovarajuću lokaciju, nakon čega oni postaju dostupni. Ovakva funkcionalnost se postiže posmatranjem direktorijuma za plugin-e upotrebom *FileSystemWatcher* klase i dinamičkim učitavanjem *assembly.-a (.dll-a)* [3]. Kada se doda nova kompajlirana klasa koja ispunjava odgovarajući interfejs, aplikacija dinamički dodaje njen sadržaj u glavni *application domain* i njene funkcionalnosti postaju dostupne.



Slika 3. Tekst profil prikaza grafa

#### 4. ZAKLJUČAK

Softversko rešenje omogućuje jednostavnu integraciju prikaza grafa u sklopu C# XAML aplikacije. Dovoljno je instancirati platno u XML delu koda i proslediti model grafa. Pošto se softversko rešenje oslanja na XML model grafa i pošto se svaki objektni model može serijalizovati u vidu XML datoteke moguće je prikazati bilo koji objektno modelovani sistem.

Kvalitet prikaza grafa uveliko zavisi od dobro napravljenog modela, profila i odabira odgovarajućeg algoritma. Ukoliko ponuđeni algoritmi ne odgovaraju, softversko rešenje omogućuje proširenje i integraciju sa drugim bibliotekama kao što su *GraphViz* i *yEd*.

Pošto se elementi iscrtavaju XAML opisima, moguće je napraviti prikaze sa mešovitim i raznolikom grafikom u vidu rasterskih i vektorskih slika i oblika.

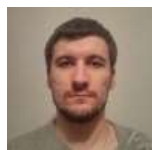
Softversko rešenje ostavlja mesta za unapređenje u vidu performansi. Nije moguće prikazati elemente realnog sistema čija brojnost prevazilazi 10 000. Komponenta za stvaranje modela grafa i njegovo raspoređivanje je ta koja je najviše usporena. Pored performansi potrebno je proširiti XAML instanciranje komponente registovanjem specijalizovanih atributa. Primeri takvih atributa bili bi vezani za podešavanja parametara algoritama za raspoređivanje u cilju dobijanja što kvalitetnijeg i lepšeg prikaza grafa. Primeri navedenih parametara su: minimalno rastojanje elemenata, širina i visina elementa.

Pored navedenog, potrebno je i implementirati funkcionalnost ocenjivanja prikazanog grafa. Ocenjivanje grafa se može izvršiti na osnovu broja čvorova i ivica koji se preklapaju ili presecaju.

#### 5. LITERATURA

- [1] <https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.shapes> (pristupljeno u septembru 2018.)
- [2] <https://github.com/Microsoft/automatic-graph-layout> (pristupljeno u septembru 2018.)
- [3] <https://docs.microsoft.com/en-us/dotnet/api/system.io.filesystemwatcher?view=netframework-4.7.2> (pristupljeno u septembru 2018.)

#### Kratka biografija:



**Momir Marić** rođen je u Travniku 1990. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva odbranio je 2018.god.

kontakt: maric.momir@gmail.com