

Graphviz



Narzędzia programistyczne

Piotr Warzocha

Plan prezentacji

- ❑ Słów kilka o Graphvizie
- ❑ Narzędzia Graphviza
- ❑ Aplikacje wykorzystujące Graphviza
- ❑ Język DOT
- ❑ Przykłady wykorzystania Graphviza
- ❑ Graphviz a .Net, Python
- ❑ Graphviz jako biblioteka

Słów kilka o Graphvizie

- ❑ <http://www.graphviz.org/>
- ❑ Graphviz – Graph Visualization Software
- ❑ Zestaw narzędzi open source do wizualizacji grafów.
- ❑ Wykorzystuje język programowania DOT do definiowania diagramów.
- ❑ Możliwość zapisu diagramów do formatów min: JPG, PNG, Postscript, PDF, SVG.
- ❑ Możliwości Graphviza:
<http://www.graphviz.org/Gallery.php>

Narzędzia Graphviza

- ❑ dot – hierarchiczne rysowanie skierowanych grafów (Sugiyama-style graph drawing - http://en.wikipedia.org/wiki/Layered_graph_drawing)
- ❑ neato – „spring model” – nie wiadomo czego spodziewać się po strukturze grafu, względnie mała liczba wierzchołków, minimalizacja globalnej funkcji energii
- ❑ fdp – „spring model”, podobny do neato
- ❑ sfdp – multiscale fdp
- ❑ twopi – wierzchołki umieszczone na współśrodkowych okręgach, odległość zależna od dystansu od wierzchołka root
- ❑ circo – grafy w postaci okręgów

Narzędzia Graphviza

- ❑ Dotty – narzędzie do tworzenia i wizualizacji grafów.
- ❑ WebDot – skrypt CGI umożliwiający osadzanie grafów w dokumentach HTML
(<http://www.graphviz.org/webdot/index.html>)
- ❑ Graphviz w przeglądarce <http://hughesbennett.co.uk/Graphviz>
- ❑ Niezliczona ilość edytorów z własnym GUI, webowe interfejsy.

Aplikacje wykorzystujące Graphviza

- ❑ ArgoUML
- ❑ Doxygen
- ❑ GraphViz jako rozszerzenie MediaWiki
<http://www.mediawiki.org/wiki/Extension:GraphViz>
- ❑ QuickGraph
- ❑ Sphinx
- ❑ Trac - <http://trac-hacks.org/wiki/GraphvizPlugin>
- ❑ I wiele więcej: <http://en.wikipedia.org/wiki/Graphviz>

Język DOT

- ❑ <http://www.graphviz.org/Documentation.php>

- ❑ Przykład polecenia:

```
dot -Tps graph1.dot -o graph1.ps
```

- ❑ Prosta składnia.

- ❑ Przykłady.

Przykłady wykorzystania Graphviza

- Wizualizacja wywołań funkcji

<https://www.ibm.com/developerworks/linux/library/l-graphvis/>

Graphviz a .NET, Python

- ▣ Python - Pygraphviz

<http://networkx.lanl.gov/pygraphviz/>

Tworzenie, edytowanie, renderowanie grafów.

Przykłady.

- ▣ .NET - QuickGraph

<http://quickgraph.codeplex.com/>

Abstrakcyjne struktury danych, implementacja grafowych algorytmów, możliwość wygenerowania kodu w języku DOT.

.NETowa biblioteka do rysowania grafów –niezwiązana z Graphvizem <http://research.microsoft.com/en-us/projects/msagl/>

Graphviz jako biblioteka

- ▣ C/C++
- ▣ Schemat użycia:

```
Agraph_t* G;  
GVC_t* gvc;  
gvc = gvContext();           /* library function */  
G = createGraph ();  
gvLayout (gvc, G, "dot");    /* library function */  
drawGraph (G);  
gvFreeLayout(gvc, g);        /* library function */  
agclose (G);                  /* library function */  
gvFreeContext(gvc);
```

Graphviz jako biblioteka

❑ Tworzenie grafu

```
char* cp;      //DOT representation of graph  
Agraph_t* G = agmread(cp);
```

```
char* name;    //name of graph  
int type;      //type of graph: strict, directed etc  
Agraph_t* G = agopen(name, type);
```

Typy grafów:

- ❑ AGRAPH Non-strict, undirected graph
- ❑ AGRAPHSTRICT Strict, undirected graph
- ❑ AGDIGRAPH Non-strict, directed graph
- ❑ AGDIGRAPHSTRICT Strict, directed graph

Graphviz jako biblioteka

□ Tworzenie wierzchołków, krawędzi

//graph with nodes or edges, name of node

```
Agnode_t *agnode(Agraph_t*, char*);
```

//graph with edges, first node, second node

```
Agedge_t *agedge(Agraph_t*, Agnode_t*, Agnode_t*);
```

□ Podgraf

```
Agraph_t *agsubg(Agraph_t*, char*);
```

Graphviz jako biblioteka

▣ Ustawianie atrybutów

//name of component, attribute name, attribute value

agset (void*, char*, char*);

//example for node np

agset (np, "color", "blue");

Graphviz jako biblioteka

- Layout – rozmieszczenie wierzchołków, pozycje, itp.

```
gvLayout (gvc, G, "dot");  
gvLayout (gvc, G, "nop");  
gvFreeLayout(gvc, G);
```

- Renderowanie

```
gvRender (GVC_t *gvc, Agraph_t* g, char *format, FILE  
          *out);  
gvRenderFilename (GVC_t *gvc, Agraph_t* g, char *format,  
                  char *filename);
```

Koniec



Dziękuję za uwagę