



JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v
JSONu

Načrti

Viri

JSON in analiza omrežij

3. del: Opis omrežij v obliki JSON

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

1257. Sredin seminar
Ljubljana, 6. januar 2016

JSON

V. Batagelj

Izhodišča

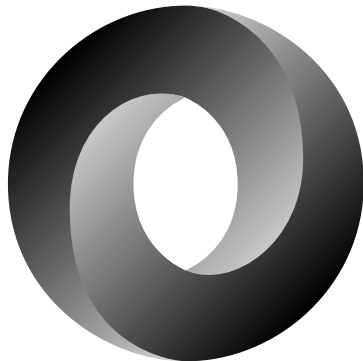
Omrežja

Opis omrežij v JSONu

Načrti

Viri

- 1 Izhodišča
- 2 Omrežja
- 3 Opis omrežij v JSONu
- 4 Načrti
- 5 Viri



Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Zadnja različica teh prosojnic (7. januar 2016 ob 06:18):

<http://vlado.fmf.uni-lj.si/seminar/06jan16/sreda1257.pdf>

- **cilj 1:** ponuditi uporabnikom obliko zapisa, ki omogoča razmeroma preprosto izmenjavo omrežnih podatkov med različnimi programi.
- **cilj 2:** ponuditi obliko zapisa, ki omogoča tudi opise prikazov omrežij in zanjo razviti ustrezno programsko podporo. Pravtako za pripravo in vzdrževanje omrežnih podatkov.
- **prikazi:** iz velikih omrežij izrežemo manjša obvladljiva podomrežja in jih dopolnimo s prikaznimi podatki. Te prikazovalniki uporabijo pri pripravi različnih prikazov.
- **“polnost” opisa:** zapis omogoča tudi opis večrelacijskih, časovnih in dvovrstnih ter zbirk omrežij.
- **dodatki:** vključevanje podatkov z datotek (import), kljukice (hooks, ports) – mesta pripenjanja povezav na like v vozliščih.

Omrežje $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ je določeno z:

- **grafom** $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, kjer je \mathcal{V} množica vozlišč, \mathcal{A} je množica usmerjenih povezav, in \mathcal{E} je množica neusmerjenih povezav. Z $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$ označimo množico vseh povezav.
 $n = |\mathcal{V}|$, $m = |\mathcal{L}|$
- \mathcal{P} **lastnosti** / vozliščne vrednosti: $p : \mathcal{V} \rightarrow A$
- \mathcal{W} **uteži** / povezavne vrednosti: $w : \mathcal{L} \rightarrow B$

JSON

V. Batagelj

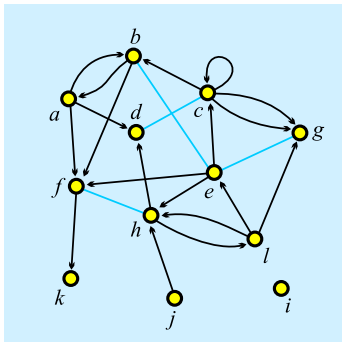
Izhodišča

Omrežja

Opis omrežij v JSONu

Načrti

Viri



$$\mathcal{V} = \{a, b, c, d, e, f, g, h, i, j, k, l\}$$

$$\mathcal{A} = \{(a, b), (a, d), (a, f), (b, a), (b, f), (c, b), (c, c), (c, g), (c, g), (e, c), (e, f), (e, h), (f, k), (h, d), (h, l), (j, h), (l, e), (l, g), (l, h)\}$$

$$\mathcal{E} = \{(b: e), (c: d), (e: g), (f: h)\}$$

$$\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{E})$$

$$\mathcal{L} = \mathcal{A} \cup \mathcal{E}$$

$\mathcal{A} = \emptyset$ – *neusmerjen* graf; $\mathcal{E} = \emptyset$ – *usmerjen* graf.

Pajek: [GraphSet](#); [TinaSet](#);

WWW: [GraphSet / net](#); [TinaSet / net](#), slika [picture](#).

JSON

V. Batagelj

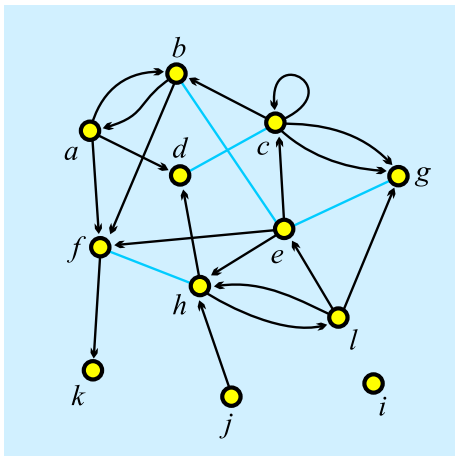
Izhodišča

Omrežja

Opis omrežij v JSONu

Načrti

Viri



```

*Vertices 12
1 "a" 0.1020 0.3226
2 "b" 0.2860 0.0876
3 "c" 0.5322 0.2304
4 "d" 0.3259 0.3917
5 "e" 0.5543 0.4770
6 "f" 0.1552 0.6406
7 "g" 0.8293 0.3249
8 "h" 0.4479 0.6866
9 "i" 0.8204 0.8203
10 "j" 0.4789 0.9055
11 "k" 0.1175 0.9032
12 "l" 0.7095 0.6475
*Arcs
1 2
2 1
1 4
1 6
2 6
3 2
3 3
3 7
3 7
5 3
5 6
5 8
6 11
8 4
10 8
12 5
12 7
8 12
12 8
*Edges
2 5
3 4
5 7
6 8
    
```

Časovno omrežje

$$\mathcal{N}_T = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W}, T)$$

dobimo, če navadnemu omrežju dodamo **čas** T . T je množica **časovnih točk** ali **trenutkov** $t \in T$.

V časovnem omrežju vozlišča $v \in \mathcal{V}$ in povezave $l \in \mathcal{L}$ niso nujno vseskozi dejavna/prisotna. Če je povezava $l(u, v)$ dejavna v trenutku t , morata biti dejavni v t tudi njeni krajišči u in v .

Omrežje sestavljeno iz vozlišč in povezav dejavnih v trenutku $t \in T$ bomo označevali z $\mathcal{N}(t)$ in mu rekli **časovna rezina** v trenutku t . V programu Pajek dobimo zaporedje časovnih rezin z zahtevo

Network/Temporal Network/Generate in time

```
*Vertices 3
1 "a" [5-10,12-14]
2 "b" [1-3,7]
3 "e" [4-*]
*Edges
1 2 1 [7]
1 3 1 [6-8]
```

Podatke o obdobjih prisotnosti/ dejavnosti posameznega vozlišča/ povezave navedemo na koncu pripadajoče vrstice v oglatih oklepajih [in]. Obdobja so oštevilčena od 1 naprej. Pri naštevanju jih ločimo z vejico ,. Zaporedna obdobja od začetka z do konca k lahko krajše zapišemo $z-k$. Znak * pomeni neskončno. Vozlišče a je prisotno v obdobjih 5 do 10 in 12 do 14. Povezava (1 : 3) je prisotna v obdobjih 6 do 8.

Povezava je prisotna, če sta prisotni obe njeni krajišči.

Time.net.

Opis časovnih omrežij uporabljen v programu Pajek je omejen – spreminjajo se lahko tudi vrednosti lastnosti in uteži. V razvoju je poseben program, ki bo omogočal opis in analizo splošnejših časovnih omrežij.

Event	Explanation
TI t	initial events – following events happen when time point t starts
TE t	end events – following events happen when time point t is finished
AV vns	add vertex v with label n and properties s
HV v	hide vertex v
SV v	show vertex v
DV v	delete vertex v
AA uvs	add arc (u,v) with properties s
HA uv	hide arc (u,v)
SA uv	show arc (u,v)
DA uv	delete arc (u,v)
AE uvs	add edge $(u:v)$ with properties s
HE uv	hide edge $(u:v)$
SE uv	show edge $(u:v)$
DE uv	delete edge $(u:v)$
CV vs	change property of vertex v to s
CA uvs	change property of arc (u,v) to s
CE uvs	change property of edge $(u:v)$ to s
CT uv	change (un)directedness of line (u,v)
CD uv	change direction of arc (u,v)
PE uvs	replace pair of arcs (u,v) and (v,u) by single edge $(u:v)$ with properties s
AP uvs	add pair of arcs (u,v) and (v,u) with properties s
DP uv	delete pair of arcs (u,v) and (v,u)
EP uvs	replace edge $(u:v)$ by pair of arcs (u,v) and (v,u) with properties s

s je lahko prazen.

Pri vzporednih povezavah $:k$ označuje k -to povezavo – HE:3 14 37
skrije tretjo neusmerjeno povezavo med vozliščema 14 in 37.

*Vertices 3

*Events

TI 1
AV 2 "b"
TE 3
HV 2
TI 4
AV 3 "e"
TI 5
AV 1 "a"
TI 6
AE 1 3 1
TI 7
SV 2
AE 1 2 1
TE 7
DE 1 2
DV 2
TE 8
DE 1 3
TE 10
HV 1
TI 12
SV 1
TE 14
DV 1

Time.tim. Friends.tim.

V razvoju je nov pristop k časovnim omrežjem, ki temelji na računanju s časovnimi količinami:

```
*vertices 6
1 "1"
2 "2"
3 "3"
4 "4"
5 "5"
6 "6"
*arcs
1 2 [(1, 8, 2), (9, 10, 4)]
2 3 [(2, 7, 7), (8, 10, 3)]
2 6 [(4, 6, 4), (8, 10, 5)]
3 2 [(4, 9, 5)]
3 4 [(1, 5, 5), (5, 9, 7)]
4 5 [(1, 3, 4), (3, 10, 2)]
5 1 [(3, 8, 1), (8, 9, 5)]
5 6 [(1, 5, 6), (5, 7, 3), (9, 10, 5)]
6 3 [(4, 8, 1), (8, 9, 5)]
6 4 [(3, 7, 9), (8, 10, 8)]
```

$$W[1][2]+W[2][3] = [(1, 2, 2), (2, 7, 9), (7, 8, 2), (8, 9, 3), (9, 10, 7)]$$

$$W[1][2]*W[2][3] = [(2, 7, 14), (9, 10, 12)]$$

Večrelacijsko omrežje sestavlja več relacij $\mathcal{L} = (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k)$ na isti množici vozlišč.

Na primer: omrežje avtobusnih postaj v mestu in avtobusnih prog, omrežje odnosov med besedami **WordNet** (synonymy, antonymy, hyponymy, meronymy, ...), omrežja odnosov med državami **KEDS** (Visit, Ask information, Warn, Expel person, ...), ...

Pri pomenski analizi besedil so enote (največkrat enostavni stavki) kodirane po shemi o-P-p (*osebek-Povedek-predmet*) ali njenih izpeljankah.



Primeri: **Roberto Franzosi**; **KEDS**, **Tabari**.

To kodiranje določa večrelacijsko omrežje na množici vozlišč $Osebk_i \cup Predmet_i$ s povezavami iz relacij *Povedki*.

Glejte tudi **RDF** v **pomenskem spletu** (semantic web), **SPARQL**.



... Večrelacijska omrežja

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v
JSONu

Načrti

Viri

To lahko naredimo na dva načina:

- geslom, ki napovedujejo opis povezav, (`*arcs`, `*edges`, `*arcslist`, `*edgeslist`, `*matrix`) dodamo številko relacije in lahko tudi njeno ime. Npr.

```
*arcslist :3 "posojanje gradiv"
```

Vse geslu podrejene povezave pripadajo navedeni relaciji. ([Sampson](#), [SampsonL](#))

- Med povezavami, podrejenimi gesloma `*arcs` ali `*edges`, lahko posamezno povezavo pripišemo izbrani relaciji, tako da njen opis začnemo s številko relacije

```
3: 47 14 5
```

Povezava s krajiščema 47 in 14 ter utežjo 5 pripada relaciji 3.



Večrelacijsko časovno omrežje – KEDS/WEIS

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v JSONu

Načrti

Viri

```

% Recoded by WEISmonths, Sun Nov 28 21:57:00 2004
% from http://www.ku.edu/~keds/data.dir/balk.html
*vertices 325
1 "AFG" [1-*]
2 "AFR" [1-*]
3 "ALB" [1-*]
4 "ALBMED" [1-*]
5 "ALG" [1-*]
...
318 "YUGGOV" [1-*]
319 "YUGMAC" [1-*]
320 "YUGMED" [1-*]
321 "YUGMTN" [1-*]
322 "YUGSER" [1-*]
323 "ZAI" [1-*]
324 "ZAM" [1-*]
325 "ZIM" [1-*]
*arcs :0 "*** ABANDONED"
*arcs :10 "YIELD"
*arcs :11 "SURRENDER"
*arcs :12 "RETREAT"
...
*arcs :223 "MIL ENGAGEMENT"
*arcs :224 "RIOT"
*arcs :225 "ASSASSINATE TORTURE"
*arcs
224: 314 153 1 [4]      890402 YUG KSV 224 (RIOT) RIOT-TORN
212: 314 83 1 [4]      890404 YUG ETHALB 212 (ARREST PERSON) ALB ETHNIC JAILED
224: 3 83 1 [4]        890407 ALB ETHALB 224 (RIOT) RIOTS
123: 83 153 1 [4]      890408 ETHALB KSV 123 (INVESTIGATE) PROBING
...
42: 105 63 1 [175]    030731 GER CYP 042 (ENDORSE) GAVE SUPPORT
212: 295 35 1 [175]   030731 UNWCT BOSSER 212 (ARREST PERSON) SENTENCED TO PRIS
43: 306 87 1 [175]    030731 VAT EUR 043 (RALLY) RALLIED
13: 295 35 1 [175]    030731 UNWCT BOSSER 013 (RETRACT) CLEARED
121: 295 22 1 [175]   030731 UNWCT BAL 121 (CRITICIZE) CHARGES
122: 246 295 1 [175]  030731 SER UNWCT 122 (DENIGRATE) TESTIFIED
121: 35 295 1 [175]   030731 BOSSER UNWCT 121 (CRITICIZE) ACCUSED

```

V *dvovrstnem* omrežju $\mathcal{N} = ((\mathcal{V}_1, \mathcal{V}_2), \mathcal{L}, \mathcal{P}, \mathcal{W})$ je množica točk sestavljena iz dveh ločenih množic vozlišč \mathcal{V}_1 in \mathcal{V}_2 , $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, povezave iz množice \mathcal{L} pa imajo eno krajišče v \mathcal{V}_1 drugo pa v \mathcal{V}_2 . Običajno je znana tudi *utež* $w : \mathcal{L} \rightarrow \mathbb{R} \in \mathcal{W}$; če ni, privzamemo $w(u, v) = 1$ za vse povezave $(u, v) \in \mathcal{L}$.

Dvovrstno omrežje lahko opišemo tudi s pravokotno matriko

$$\mathbf{A} = [a_{uv}]_{\mathcal{V}_1 \times \mathcal{V}_2}.$$

$$a_{uv} = \begin{cases} w_{uv} & (u, v) \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}$$

Primeri: (članki, avtorji), (ljudje, društva, leta članstva), (kupci, dobrine, količina), (poslanci, vprašanje, pozitivni glas), (ljudje, revije, branost).

Dvovrstno omrežje napovemo z *vertices $n \ n_u$.

Avtorji in dela.

ORA/CASOS meta-matrix (Agent, Knowledge, Task, Resource) [Manual](#)

Sean F. Everton: Disrupting Dark Networks. [Table 3.1 / page 69](#)

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v JSONu

Načrti

Viri



Najbolj znan primer dvovrstnega omrežja so Davisove južnakinje.

Davis.paj. Freemanov pregled.

	CODE NUMBERS AND DATES OF SOCIAL EVENTS REPORTED IN <i>Om City Herald</i>													
	(1) 6/27	(2) 3/3	(3) 4/12	(4) 9/16	(5) 2/25	(6) 5/19	(7) 3/15	(8) 9/16	(9) 4/6	(10) 6/10	(11) 2/23	(12) 4/7	(13) 11/21	(14) 8/3
1. Mrs. Evelyn Jefferson.....	X	X	X	X	X	X	X	X	X					
2. Miss Laura Mandeville.....	X	X	X	X	X	X	X	X	X					
3. Miss Theresa Anderson.....		X	X	X	X	X	X	X	X					
4. Miss Brenda Rogers.....	X		X	X	X	X	X	X						
5. Miss Charlotte McDowd.....			X	X	X	X	X							
6. Miss Frances Anderson.....			X	X	X	X	X							
7. Miss Eleanor Nye.....				X	X	X	X							
8. Miss Pearl Oglethorpe.....				X	X	X	X							
9. Miss Ruth DeSand.....				X	X	X	X							
10. Miss Verne Sanderson.....						X	X	X	X			X		
11. Miss Myra Liddell.....							X	X	X	X				
12. Miss Katherine Rogers.....							X	X	X	X		X	X	X
13. Mrs. Sylvia Avondale.....							X	X	X	X		X	X	X
14. Mrs. Nora Fayette.....						X	X	X	X	X	X	X	X	X
15. Mrs. Helen Lloyd.....						X	X	X	X	X	X	X		
16. Mrs. Dorothy Murchison.....							X	X	X	X				
17. Mrs. Olivia Carleton.....										X	X			
18. Mrs. Flora Price.....									X		X			



Opis omrežij v JSONu

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v
JSONu

Načrti

Viri

Preprost opis omrežja, ki ga najdemo v obstoječih rešitvah v JSONu, ni najbolj primeren za opis zbirk omrežij – predstavimo jih kot več omrežij. Pri tem se več podatkov ponavlja – kar bi lahko rešili z nadaljnjim drobljenjem omrežij in vključevanjem datotek.

Drugo vprašanje je zapis posameznih sestavin (vozlišča, skupine, relacije, ...) v opisu. Obstajata dva prevladujoča pristopa: z imeni ali z indeksi. Za večja omrežja se praviloma uporablja indeksni pristop. Zanj sem se odločil tudi sam – vedno je mogoče dodati način opisa, ki podpira tudi imenski način.

Način opisa bomo določili na začetku z lastnostjo "mode", ki ima trenutno možni vrednosti "basic" in "general". Privzeta vrednost je "basic". Kasneje morda še "condensed" in "extended".

Način "basic" je namenjen opisu enostavnih omrežij na isti množici vozlišč. V bistvu sledi uveljavljeni shemi.

```
{ "network": "class",
  "info":{ "org": 1, "nNodes": 15, "nArcs" : 30, "nEdges": 13,
    "nWeak" : 1,
    "title" : "borrowing study materials",
    "meta" : { "date": "October 2015", "author": "V. Batagelj" }
  },
  "nodes": [
    { "id": 1, "short": "m02", "x": 0.1857, "y": 0.2781, "size": 1 },
    { "id": 2, "short": "m03", "x": 0.5482, "y": 0.6169, "size": 1 },
    { "id": 3, "short": "w07", "x": 0.2219, "y": 0.4526, "size": 2 },
    { "id": 4, "short": "w09", "x": 0.8078, "y": 0.3223, "size": 2 },
    ...
    { "id": 14, "short": "m89", "x": 0.4000, "y": 0.8469, "size": 1 },
    { "id": 15, "short": "m96", "x": 0.3482, "y": 0.1778, "size": 1 }
  ],
  "links": [
    { "type": "arc", "source": 6, "target": 15, "weight": 1 },
    { "type": "arc", "source": 2, "target": 7, "weight": 1 },
    ...
    { "type": "arc", "source":15, "target": 3, "weight": 1 },
    { "type": "edge", "source": 6, "target": 12, "weight": 1 },
    ...
    { "type": "edge", "source": 4, "target": 12, "weight": 1 },
    { "type": "edge", "source": 6, "target": 13, "weight": 1 }
  ]
}
```

V načinu "general" v opisu omrežja lahko nastopi več množic vozlišč, več relacij, itd. Zato jih je potrebno poimenovati in vpeljati posamezne zvrsti "type" podatkov: "nodes", "links", "vector", "permutation", "partition", "layout" in morda še kakšno.

Opis celotnega omrežja in posameznih množic je sestavljen iz delov:

"info" – parametri in izračunani podatki

"meta" – metapodatki

"default" – privzete vrednosti

"data" – (pravi) podatki

"style" – prikazna določila.

Morda je del "default" odveč in se da porazdeliti po ostalih.



Način "general"

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v
JSONu

Načrti

Viri

```
{ "mode": "general",
  "meta": [ [date1, desc1], [date2, desc2] ],
  "default": { "layout": {"coord": "circular"},
              "style": {"nodes": {"shape": "circle", "r": 1, "color": "red"},
                       "links": {"line": "solid", "lw": 2, "color": "blue"}
              },
  "links": {"linktype": "arc"}
},
"A": { "type": "nodes",
      "info": {"size": 35},
      "style": { "color": "red", "shape": "circle", "size": 5}, // def
      "data": [ {"id": 1, "short": "lab1", ... }
                ],
      ...
    },
"B": { "type": "nodes",
      ...
    },
}
```



Način "general"

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v
JSONu

Načrti

Viri

```
"R": { "type":"links",
      "nodes": ["A","B"],
      "linktype": "edge",
      "data": [{ "type": "arc", "id":1, "n1":a, "n2":b, "w":v, ...},
               ...
            ]
    },
"v": { "type":"vector",
      "info": { "size": 35, "type": "integer" },
      "nodes": "A",
      "data": [3,7,5,1,8,2,....]
    },
"ġ2": { "type": "layout",
      "nodes": ["A","B"], "links":"R",
      "assign": { "coord": ["v1","v2"], "r":"v3", "color": "p1", ...
      "style": { "shape": "square" }
    }
}
```



Način "general" – časovne količine, legende

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v
JSONu

Načrti

Viri

```
"info": { "TQ": {"first":F, "last":L} }  
"TQv" : [[s1,f1,v1], [s2,f2,v2],...]  
  
"codeOrg": 1,  
"code" : [ "Mon", "Tue", ... ]  
  
"code" : [ [2, "zadostno"], [5, "odlično"], ... ]
```

graphA.js

```
graph = { "network": "testA", "info": {"org": 0},
  "nodes": [
    { "short": "Ann", "x": 0.2, "y": 0.2, "num": 1, "Size": 100 },
    { "short": "Ben", "x": 0.2, "y": 0.8, "num": 4, "Size": 500 },
    { "short": "Tim", "x": 0.8, "y": 0.2, "num": 2, "Size": 200 },
    { "short": "Zoe", "x": 0.8, "y": 0.8, "num": 3, "Size": 400 }
  ],
  "links": [
    { "source": 0, "target": 1, "Count": 1, "Weight": 100 },
    { "source": 1, "target": 2, "Count": 1, "Weight": 100 },
    { "source": 2, "target": 3, "Count": 1, "Weight": 100 },
    { "source": 1, "target": 3, "Count": 2, "Weight": 300 }
  ]
}
```

graphB.js

```
graph = { "mode": "general", "network": "testB", "info": {"org": 0},
  "persons": { "type": "nodes", "data": [
    { "short": "Ann", "x": 0.2, "y": 0.2, "Num": 1, "Size": 100 },
    { "short": "Ben", "x": 0.2, "y": 0.8, "Num": 4, "Size": 500 },
    { "short": "Tim", "x": 0.8, "y": 0.2, "Num": 2, "Size": 200 },
    { "short": "Zoe", "x": 0.8, "y": 0.8, "Num": 3, "Size": 400 }
  ]},
  "friend": { "type": "links", "data": [
    { "source": 0, "target": 1, "Count": 1, "Weight": 100 },
    { "source": 1, "target": 2, "Count": 1, "Weight": 100 },
    { "source": 2, "target": 3, "Count": 1, "Weight": 100 },
    { "source": 1, "target": 3, "Count": 2, "Weight": 300 }
  ]}
}
```



Primer – prikaz

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v
JSONu

Načrti

Viri

graphXYd.html

```
<!DOCTYPE html>
<meta charset="utf-8">
<body>
<!-- script src="./graphA.js"></script -->
<!-- script src="./graphB.js"></script -->
<script src="./class.js"></script>
<script src="http://d3js.org/d3.v3.min.js"></script>
<script>
function drawNet(V,L){
// set up the drawing area
var width = 500,
    height = 500;
var svg = d3.select("body").append("svg")
...
    .attr("font-family", "Arial Unicode MS")
    .attr("dy", "0.5em")
    .attr("font-size", function(d) {return (20); })
    .text(function(d) { return d.icon; });
return 1
}
s = graph.info.org;
if(graph.mode=='general') {
    var OK = drawNet(graph.persons.data,graph.friend.data)
} else {
    var OK = drawNet(graph.nodes,graph.links)
}
</script>
</body>
```

class.js

JSON

V. Batagelj

Izhodišča

Omrežja







Opis omrežij v
JSONu

Načrti

Viri

- "persons": {"type": "nodes", "import": URL}
- "coord": "circular", "random"
- prikaz omrežja na ozadju (npr. zemljevid)
- recept za shranjevanje slike SVG iz spletnega pregledovalnika
- uporaba ikonskih in/ali pisav SVG za prikaz vozlišč
- izbira gesel (source, tail, nodeA, n1, from), (target, head, nodeB, n2, to)

- Preoblikovati nekaj omrežij v predlagano obliko opisa – preveriti ali so še kje težave.
- Izdelava programske podpore za zbiranje in vzdrževanje omrežij. Izvoz v izbrane oblike zapisa.
- Izdelava programske podpore za prikaze omrežij.
- Pripraviti formalni opis oblike opisa (slovnica opisa – JSON Schema).
- Predstavitev predloga (priročnik, konference, delavnice, članki).

-  Anthony Bargnesi, Anselmo DiFabio, William Hayes: JSON Graph. [GitHub](#)
-  Vladimir Batagelj: Complex Networks, Visualization of. R.A. Meyers, ed., Encyclopedia of Complexity and Systems Science, Springer 2009: 1253-1268.
-  Vladimir Batagelj, Andrej Mrvar: [Pajek manual](#).
-  Jernej Bodlaj: Network Data File Formats. in Reda Alhajj, Jon Rokne (eds.): Encyclopedia of Social Network Analysis and Mining. Springer, New York, 2014, p. 1076-1091.
-  Richard Brath, David Jonker: Graph Analysis and Visualization: Discovering Business Opportunity in Linked Data. John Wiley & Sons, Indianapolis, Indiana, 2015.
-  Emden Gansner, Eleftherios Koutsofios, Stephen North: [Drawing graphs with dot](#), January 26, 2006

JSON

V. Batagelj

Izhodišča

Omrežja

Opis omrežij v
JSONu

Načrti

Viri



Michael Himsolt: GML: A portable Graph File Format. [report](#)



Wouter De Nooy, Andrej Mrvar, Vladimir Batagelj: Exploratory Social Network Analysis with Pajek; Revised and Expanded Second Edition. Structural Analysis in the Social Sciences, Cambridge University Press, September 2011.



Matthew Roughan, Jonathan Tuke: Unravelling Graph-Exchange File Formats. [arXiv:1503.02781](#), submitted on 10 Mar 2015.



Wikipedia: [JSON](#)



Leland Wilkinson: The Grammar of Graphics. Springer-Verlag, New York, 2005.