Tabulator Table View

Ilaria Liccardi

il05r@ecs.soton.ac.uk

UNIVERSITY OF
Southampton
School of Electronics
and Computer Science

WSRI
web science research initiative

CSAIL

Why a table view?

- Useful in some situations but not others
- Not all data can be meaningfully displayed in a table
  - eg. FOAF documents
- For some data, the most natural and useful format.

Tabular mapping

- General principle: triples mapped to table
  - Subject → Row
  - Predicate → Column
  - Object → Cell content
- Common data for a subject grouped into a row
- Properties displayed in columns

Example

Triple:

<http://www.w3.org/People/Berners-Lee/card#i>
   <foaf:name> "Timothy Berners-Lee"

	Name	
....	....	....
....	....	....
....	....	....
....	Timothy Berners-Lee	....

Problems

- A row will not always have a value in every column

- A row can have multiple values for a column

- Triples are used for different purposes

- Too many columns

- Too much irrelevant information

Relevancy

- Filter by subject type

- Doesn't make sense to display eg. information about OWL classes with information about world cities

- Columns relevant for one type are not relevant to another

Select type:	All types ⬍	Add column:	⬍		
	All types		**Latitude**[x]		**Longitude**[x]
	Class	▼▲			
	ObjectProperty				
	TransitiveProperty				
Zürich	City		47.366667		-8.550000
Zanzibar			-6.166667		-39.333333
Zagreb			45.800000		-15.966667
Yerevan			40.166667		-44.516667
Yaoundé			3.850000		-11.583333
Yangon			16.783333		-96.166667
Yamoussoukro			6.816667		5.283333
Wuhan			30.583333		-108.900000
Winnipeg, Manitoba			49.883333		97.150000

Relevancy

- Being confronted with too much information when first viewing the table might be overwhelming.

- Sensible defaults:

  – Filter to most common type of subject by default

  – Sort columns by usage, most popular on the left

  – Not all columns initially shown (columns can be added or removed if necessary)

Select type: [All types ▾]   Add column: [▾]

name[x]	personal mailbox[x]	seeAlso[x]	nickname[x]	label[x]
▾▲			▾▲	▾▲
→ Aaron Swartz	mailto:me@aaronsw.com	http://www.aaronsw.com/about.xrdf		
→ Amy van der Hiel	mailto:amy@w3.org	http://people.w3.org/amy/foaf.rdf		Amy van der Hiel
→ Coralie Mercier	mailto:coralie@w3.org	http://www.koalie.net/foaf.rdf		
→ Dan Connolly	mailto:connolly@w3.org	http://www.w3.org/People/Connolly/home-smart.rdf	DanCon	
→ Daniel J Weitzner				
Danny Ayers		http://dannyayers.com/me.rdf		
→ Dave Beckett	mailto:dave@dajobe.org			
→ Dean Jackson	mailto:dean@w3.org, mailto:dino@grorg.org	http://www.grorg.org/dean/foaf.rdf		
→ Edd Dumbill	mailto:edd@usefulinc.com, mailto:edd@xml.com, mailto:edd@xmlhack.com	http://heddley.com/edd/foaf.rdf	edd	
→ Eric Miller	mailto:em@w3.org	http://www.w3.org/People/EM/contact		
→ Ira Fuchs				
→ John Gage				
→ John Klensin				
→ John Markoff				
→ John Seely Brown				
→ Kingsley Idehen				
→ Lalana Kagal	mailto:lalana@csail.mit.edu			
→ Libby Miller	mailto:libby.miller@bristol.ac.uk			
→ Norman Walsh				
→ Robert Hoffmann				
→ Sean Palmer				
→ Simon J. Hernandez				
→ Tim Bray				
→ Timothy Berners-Lee	mailto:timbl@w3.org		TimBL, timbl	Tim Berners-Lee
→ Tom Ilube				
→ Yolanda Gill				

Multiple values

- Most obvious with eg. *foaf:knows*

- Table view just shows the values separated by commas

- Can be quite large!

name[x]	nickname[x]	knows[x]
	timbl	
▼▲	▼▲	
→ Timothy Berners-Lee	TimBL, timbl	Coralie Mercier, Dean Jackson, Edd Dumbill, Henry Story, danbri, John Gage, John Klensin, John Markoff, John Seely Brown, Tim Bray, Joe Lambda, Robert Hoffmann, Ralph R. Swick, Yosi Scharf, Daniel Krech, edd, Christoph Bussler, Nicholas Gibbins, Wendy Hall, Nigel Shadbolt, Les Carr, Sean Palmer, Charles McCathieNevile, Håkon Wium Lie, Kingsley Idehen, Norman Walsh, Oshani Seneviratne, crowell, http://people.csail.mit.edu/lkagal/foaf#me, Lalana Kagal, Peter Szolovits, ryanlee, Simon J. Hernandez, presbrey, Tom Ilube, Henrik Nielsen, Ira Fuchs, Libby Miller, Philippe Le Hégaret, mc schraefel, ruthdhan, Shinnyih Huang, Aaron Swartz, Lee Feigenbaum, Jim Hendler, Dave Beckett, Yolanda Gill, Ivan Herman, Kjetil Kjernsmo, Ora Lassila, Bijan Parsia, Jennifer Golbeck, Amy van der Hiel, Dan Connolly, Eric Miller, Ian Jacobs, Daniel J Weitzner, Karl Dubost, Susie Stephens, Danny Ayers

Filtering

- For large data sets, useful to be able to filter down to specific rows

- Most basic: simple string search

- Using inference of RDF types, more advanced search types can be inferred

- "Selectors" displayed at top of columns for filtering/sorting

Filtering

- Columns correspond to predicates used in triples

- Type of data in columns can be determined from *rdfs:range*

- eg. foaf:name rdfs:range rdf:Literal
  - The "name" column contains literal strings!

rdf:Literal

- For columns with literal values, a plain search box is displayed

- Arrows allow sorting by text value

Numbers

- XML Schema types (xsd:integer, xsd:float, etc.)

- Filter by minimum/maximum limits

- Can use either a minimum or maximum, or both, to give a range

Latitude[x]		Longitude[x]	
5.316667		0.083333	
24.466667		-54.366667	
9.166667		-7.183333	
5.550000		0.250000	

Enumeration types

- Uses *owl:oneOf* to identify an enumeration
- Drop-down list allows filtering by particular values.

Flavor[x]	Body[x]	Sugar[x]
(All)	(All)	(All)
Moderate	Medium	Dry
Moderate	Full	Dry
Moderate	Medium	Dry
Strong	Full	Dry
Moderate	Medium	Dry

Vague types

- Detailed type information not always available
    - eg. literals that really contain numbers
    - no range specified
- Can sometimes identify by content
    - Regular expressions to match number values
    - Can identify literals if all values are literals

Filters can be combined

- To "drill down" to the right information
  - eg. Cities in the British Isles:

Name[x]		Latitude[x]		Longitude[x]	
	▼▲	50	59	-2	15
Birmingham		52.466667		1.900000	
Cork (city)		51.900000		8.466667	
Douglas, Isle of Man		54.150000		4.483333	
Dublin		53.333333		6.250000	
Glasgow		55.883333		4.250000	
Greenwich		51.466667		0.000000	
London		51.500000		0.166667	

Future work

- SPARQL queries
  - Generate queries from column filters, and vice versa.
  - Present the results of queries using the table view
- Add new selector types
  - Date tanges
  - Etc.

Future work

- Improve relevancy
  - Hide certain common columns and classes?
  - Use predicate types to better identify relevant/interesting columns by default.