# ARIA for roles and charts

Lisa Seeman - Draft 1 for brain storming

**Aim:** To represent all different charts and graphs in and machine determinable semantics, rich in meaning, the basis of application platforms and, hence, most importantly, screen reader friendly.

This does not address display information which will be addressed by the native environment such as SVG. I have messed (for bainstorming) around with different ways to represent things, so syntax will vary.

To make it easier to read, and technology agnostic I have made the roles the element names.

Eg instead of *<g role=”axis”>* I have just write *<axis>*

 Properties are …properties.

We are building charts and graphs were nodes can have meaning and values in multi dimensions, for example it can change in time as well as x and y.

Sometime a chart can support even more dimensions

Also please look at the notes at the end. They should explain the power of this model. Don’t get hung up on any semantic errors. We can iron it out if we like the model. And look at the version for lazy people at the end. It is less powerful but easier for adoption. It allows adoption to be compatible with a powerful platform for folks who want it.

This document assumes you ca read xml. It should be parsable to RDF but it is not necessary.

### Example 1:

This example is from https://www.youtube.com/watch?v=H6zg3nWXeP4 from approximately 9 minutes in. (A very cool video.)

In the chart we have, x (life expectancy) , y ( births per woman) , the values change over our next axis - time, color represent s continent and each node has a size that represents population size (changes over time). So that is 5 values or “axis” that need to be represented. Each value can change as a formula of the other axis (such as time)

Figure 1.1 and figure 1.2 - screen shots of graph at two points in time



Figure 1.3 example with line drawn for world average at 2.5 children per woman in 2012



### Example 2

A train map.

This has nodes and connectors. Both nodes and connectors can have types – such as lines names and subtypes such as wheel chair access or bathrooms

My simple example has 3 stations (Jerusalem, Beit Shemesh and Tel Aviv) and two type of lines (fast and standard) Jerusalem and Tel Aviv are big stations and Tel Aviv has bathrooms (Actually they all have bathrooms, it is just an example). There is a fast train between Jerusalem and Tel Aviv.

Figure 2 – stations and train lines described above.

## Worked examples

### Note:

This is an initial brain storm for the overall architecture and concept. Details need work, and there are to do items inside. If we like the overall architecture I can draft the todo items.. The values are just made up.

#### Example 1 - chart

Chart semantics for example 1 above

(Note I have no idea how what the plural of axis is. Used axisses….)

<chart display =”node”> (display could be pie chart, bar chart ect)

<axisses>

<axis id= “x” aria-label=” life expectancy at birth“ aria-describedby=”years” values = “40 50 60 70 80 90 100”> add markup to draw axis here ,

 give text format information

…optional add native display information color width on how to draw also bars or charts eg gradient along this axis </axis>

<axis id= “y” aria-label=” birth per women“ values = “1 2 2.2 2.5 3 4 5 5.5 6 7”> <display> …optional add native display information color width etc. Could include non linear algorithm for axis.</display></axis>

<axis id= “continent” aria-label=” continent“ values = “Asia, Africa, Europe, Americas …”> <display> <bind value=”Africa” color=”blue”> (TODO:: I am sure better binding can be made – but put binding here…)</display></axis>

OR

<axis id= “continent” aria-label=” continent“ values = “blue, green , red, yellow …” (Value can be display info?) aria-value-labels = “Asia, Africa, Europe, Americas …” class= “continentaxis” (class instead of display) />

<axis id= “year” aria-label= “year “ aria-valuemin = “1970” aria-valuemax=”2013” aria-stepsize=”1” aria-draw=”false”(used for display of population not drawn directly ) > </axis>

<axis id= “population” aria-label= “population“ aria-describedby=”increment in millions” aria-bind= “year”> <display> (circumference etc)</display></axis>

 Etc.

</ axisses >

<node aria-label= “ United states of America” value=”(79 79 80 ,,,82,82(year)) , 2.2 , Americas, (200+F(x)(year))” > and additional drawing information</node>

Todo: Work on clear way to write values per year and other multi dimensional formula, whilst allowing it to be simple coordinates for the majority of 2 dimensional charts. )

…<node aria-label= “Bangladesh” etc…>…..

<line aria-label= “world average children per woman” value=”(5.5(year=”1962”) ……2.5 (year=”2012”), NA , NA, NA, )” aria-importance = “10” aria-type =” average “> <display> draw line based on aria selector of year = “2012” or year = “1962” </display></node>

Todo: how to represent axis that are constant for a node (IE a line)

<line aria-label= “world average life span” value=”(NA, 54 56…72 (year), NA , NA, NA, )” aria-importance = “9” class= “invisible” aria-type =”average“</node>

</chart>

Note: we want this visible to the screen reader but non visible to a sited user. This is just to show what people conclude with the visual, that as world life expectancy is increasing across all the globe. From that you see the correlation of the data with decreasing birth per women.

Note: **aria-importance** could have a range 0-10 (or 0-1 etc) the point is by adding this we point out that among all the data this is what you really needed to “see” **to follow the concept and complete the task** (in this case the task is understanding that global birth per women is decreasing). The second most important task is to “get” is life expectancy increasing correlates to decreasing birth per women.

Other information really just supports this.

Note: more on use of aria-type below this will help us say what things are. Some can be standard such as “average”.

#### Example 2 - graph

Graph semantics for example 2 above.

<graph>

<node label= “station” id= “abstractstation” aria-abstract=”true” > <display> draw circle here</display></node>

<node label= “big” id= “abstractbigstation” type =“abstractstation” aria-abstract=”true” > <display> color=red ( inherits and overrides display from abstractstation) </display></node>

<node label= “with bathrooms” id= “abstractbigstationwc” type =“abstractbigstation” aria-abstract=”true” > <display> markup to draw bathroom icon hear, draw tacto grafics option?</display></node>

<node label= “Beit shemesh” type = “abstractstation” />

(same as <group role= “aria- node” label= “Beit shemesh” type = “abstractstation” > add location information to draw </group>)

Note this syntax should be rdf compatible so that I could have all my abstracts in and RDF schema and say

<node label= “Beit shemesh” id= “beitshemesh” type = “uri#abstractstation” /> were uri is the namespace reference)

<node label= “Jerusalem” id= “Jerusalem“ type = “abstractbigstation” >add native display location info </node>

<node label= “Telaviv” id= “Telaviv” type = “abstractbigstationwc” /> add native display location info </node>

…………Now for the connectors

<connector label= “train line” id= “abstractline” aria-abstract=”true” > <display> thichness info ect</display></connector>

 <connector label= “ fast” id= “abstractfastline” type= “abstractline” aria-abstract=”true” > <display> dashed or arrows drawing</display></connector>

<connector aria-start=”Jerusalem” aria-end=”beitshemesh” type= “abstractline” > <display> all drawing info is already provided – no need to do anything but can override if you want</display></connector>

<connector aria-start=”Telaviv” aria-end=”beitshemesh” type= “abstractline” > <display> all drawing info is already provided – no need to do anything but can override if you want</display></connector>

<connector aria-start=”Telaviv” aria-end=”Jerusalem” type= “abstractfastline” > <display> all drawing info is already provided – no need to do anything but can override if you want</display></connector>

Note – start and end can also be coordinates if preferred

A connector is a type of line.

</graph>

##### Example 2 - Version 2 for lazy people….

This one does not use abstract types. Less powerful but less thought is needed

<graph>

<node label= “Beit shemesh station” id= “beitshemesh” class = “station” ….add native display location info />

<node label= “Jerusalem station” class = “bigstation” id = “Jerusalem”….add native display location info />

<node label= “Tel Aviv station” class = “bigstationwc” id = “Telaviv” … add native display location info />

…………Now for the connectors

<connector aria-start=”Jerusalem” aria-end=”beitshemesh” class = “trainline1” />

<connector aria-start=”Telaviv” aria-end=”beitshemesh” class = “trainline1” />

<connector aria-start=”Telaviv” aria-end=”Jerusalem” class = “trainlinefast” />

Note – start and end can also be coordinates if preferred

</graph>

Some notes

1. I have kept the role to be the base type (node connector etc) and not the actual inherited type – to make it easier to implement without inheritance.
2. Labels and descriptions for inherited abstract types are concatenated.
3. **You do not need to use abstract types**, but you can. It makes info and work reusable. Makes it possible to write apps to run on top of graphs and hugely good stuff. And of course, you do not need to re-label (unless you want to add more). This is important because labels will be consistent and logical across large diagrams or groups of diagrams. If an item is the same thing it will have the same label. When you use a type you inherit all the class info ect.
4. **Abstract types could be in a linked-to file**
5. You can also reference an RDF taxonomy node as your type giving it rich semantic meaning and extendibility etc… We need t confirm that it is all rdf parsable and adjust to make it so.
6. Default is aria-abstract=”false”
7. Of course, if rendering information is in CSS you can use CSS selectors for rendering different types.
8. Tons of discussion points if we decide we like it. - should both examples be called graphs or charts? Etc…