

Sonification[S]

proc: Proc[S]

sources : evt.Map[S, String, Sonification.Source]

~~controls: evt.Map[S, String, DoubleObj]~~

controls: Controls[S]

attr

type Controls[S] = Folder[S] ?!

attr

"layout"

*SmallTalk FTW!*

"padding"

"background"

"visible"

Proc[S]

graph: SynthGraphObj.Var[S]

outputs: Outputs[S]

attr

- used by graph.Attribute

so a folder is a container, the containing elements are the main "models"  
and their respective attribute maps configure the UI views?

Then a default "layout" for a folder is just that - a folder tree view

A hen an egg question: does the container contain the models who carry their views  
as attribute, or vice versa. One would think that MVC implies the latter, because the  
same model can be viewed multiple times. That in turn means we need at least one  
object type for widget, possibly one per widget.

DoubleWidget:

model: DoubleObj

attr

visible

enabled

min

max

renderer: String of "slider", "rotary", etc.

BooleanWidget:

model: BooleanObj

attr

visible

enabled

renderer: String of "checkbox", "toggle-button", ...