



Kepler Exercise

Deana Pennington
University of New Mexico
January 12, 2006





Graphing data

Vegetation biomass test data

1. With EML
2. Without EML





Sevilleta LTER Biomass Plots

Biomass = $f(\text{species, cover\%, height, season})$

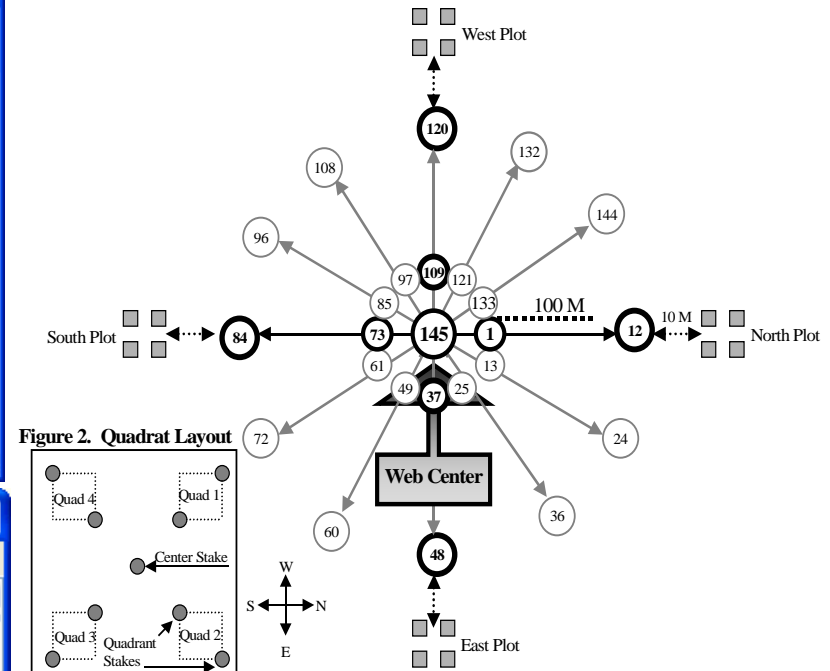
Microsoft Excel - 1999_npp.csv

	A	B	C	D	E	F	G	H	I	J	K	L
1	DATE	SITE	WEB	PLOT	QD	SPECIES	OBS	COVER	HEIGHT	COUNT	PHEN	COMMENT
2	2/3/1999	FPC	1	E	1	ERPU8	1	0.5	4	13	V	NA
3	2/3/1999	FPC	1	E	1	ERPU8	2	0.1	2	16	V	NA
4	2/3/1999	FPC	1	E	1	GUSA2	1	0.01	4	2	V	NA
5	2/3/1999	FPC	1	E	1	GUSA2	2	0.1	5	1	V	NA
6	2/3/1999	FPC	1	E	1	GUSA2	3	0.5	12	1	V	NA
7	2/3/1999	FPC	1	E	1	LEFE	1	0.25	5	1	V	NA
8	2/3/1999	FPC	1	E	2	LATR2	1	7	36	2	V	NA
9	2/3/1999	FPC	1	E	2	LATR2	2	2	32	3	V	NA
10	2/3/1999	FPC	1	E	2	LATR2	3	8	61	1	V	NA
11	2/3/1999	FPC	1	E	2	LATR2	4	3	45	1	V	NA
12	2/3/1999	FPC	1	E	2	LATR2	5	2	24	1	V	NA
13	2/3/1999	FPC	1	E	2	ERPU8	1	0.25	3	3	V	NA
14	2/3/1999	FPC	1	E	2	ERPU8	2	0.05	2	11	V	NA
15	2/3/1999	FPC	1	E	2	LEFE	1	0.1	4	3	V	NA

1999_npp.csv - Notepad

```

File Edit Format View Help
DATE, SITE, WEB, PLOT, QD, SPECIES, OBS, COVER, HEIGHT, COUNT, PHEN, COMMENTS
2/3/1999, FPC, 1, E, 1, ERPU8, 1, 0.5, 4, 13, V, NA
2/3/1999, FPC, 1, E, 1, ERPU8, 2, 0.1, 2, 16, V, NA
2/3/1999, FPC, 1, E, 1, GUSA2, 1, 0.01, 4, 2, V, NA
2/3/1999, FPC, 1, E, 1, GUSA2, 2, 0.1, 5, 1, V, NA
2/3/1999, FPC, 1, E, 1, GUSA2, 3, 0.5, 12, 1, V, NA
2/3/1999, FPC, 1, E, 1, LEFE, 1, 0.25, 5, 1, V, NA
2/3/1999, FPC, 1, E, 2, LATR2, 1, 7, 36, 2, V, NA
2/3/1999, FPC, 1, E, 2, LATR2, 2, 2, 32, 3, V, NA
2/3/1999, FPC, 1, E, 2, LATR2, 3, 8, 61, 1, V, NA
2/3/1999, FPC, 1, E, 2, LATR2, 4, 3, 45, 1, V, NA
2/3/1999, FPC, 1, E, 2, LATR2, 5, 2, 24, 1, V, NA
2/3/1999, FPC, 1, E, 2, ERPU8, 1, 0.25, 3, 3, V, NA
  
```





This Exercise

- Using a subset of the data for 1 species in 1 quadrat, extract vegetation cover% and height, and create bar graphs of those values through time

```
veg_data_test.csv - Notepad
File Edit Format View Help
DATE,SITE,WEB,PLOT,QD,SPECIES,OBS,COVER,HEIGHT,COUNT,PHEN,COMMENTS
2/3/1999,FPC,1,E,1,ERPU8,1,0.5,4,13,V,NA
2/3/1999,FPC,1,E,1,ERPU8,2,0.1,2,16,V,NA
6/2/1999,FPC,1,E,1,ERPU8,1,0.5,6,2,NA,NA
6/2/1999,FPC,1,E,1,ERPU8,2,0.25,4,12,NA,NA
6/2/1999,FPC,1,E,1,ERPU8,3,0.1,3,10,NA,NA
6/2/1999,FPC,1,E,1,ERPU8,4,0.05,2,13,NA,NA
10/7/1999,FPC,1,E,1,ERPU8,1,0.25,7,5,F,NA
10/7/1999,FPC,1,E,1,ERPU8,2,0.1,7,2,F,NA
10/7/1999,FPC,1,E,1,ERPU8,3,0.01,2,31,F,NA
```





Get data file

- Copy from
tundra://training_lab/downloads
veg_data_test.csv
vegdataDP.xml
vegdata2DP.xml
- Copy to
c:\kepler\workflows\eco





View the end result

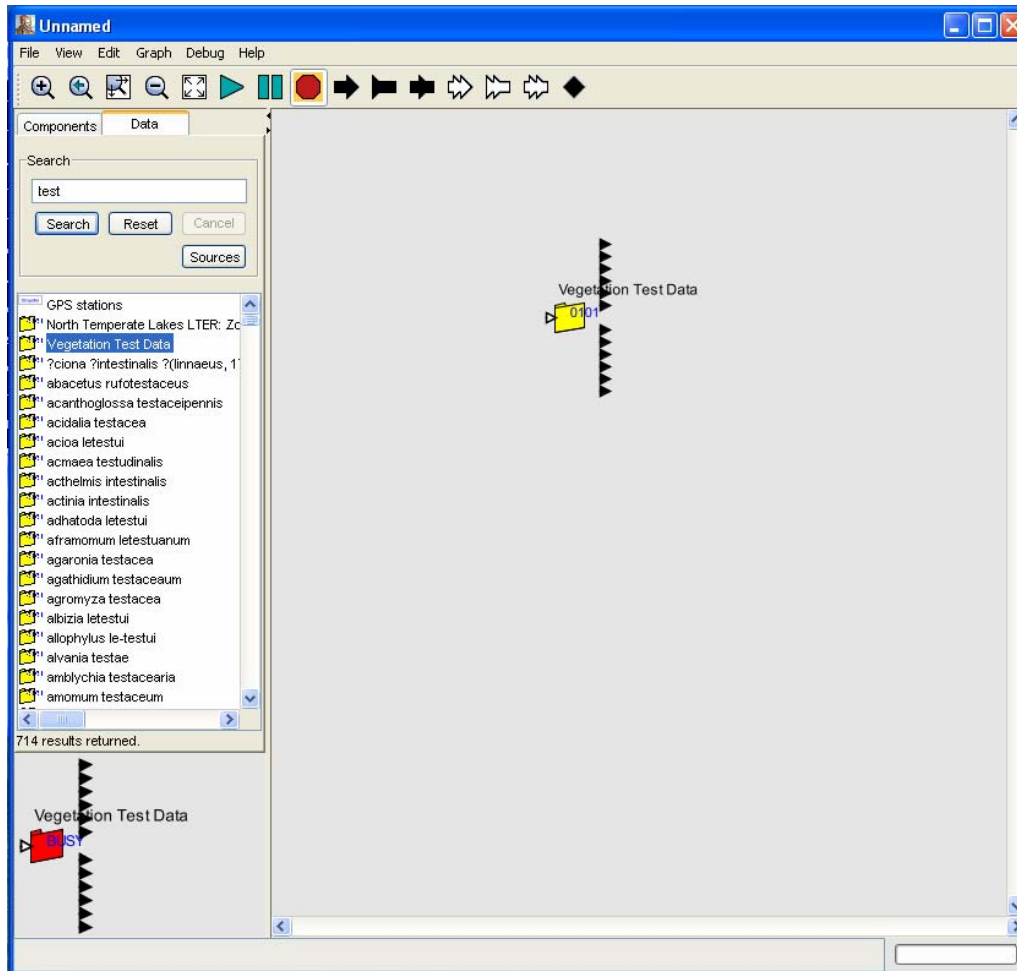
- Open vegdataDP in kepler
 - File->Open->vegdataDP.xml
- Run





With metadata (easier)

- Open a new graph editor in Kepler
File->New->Graph Editor
- Switch to the Data tab
- Remove KU Digir and Geon from Sources
- Search for "test"
- Add "Vegetation Test Data" to canvas
Drag and drop





Explore the dataset

Edit parameters for Vegetation Test Data

EML File: Browse

schemaDef:

```
<schema>
<table name="veg_data_test.csv">
  <field name="DATE" dataType="STRING"/>
  <field name="SITE" dataType="STRING"/>
  <field name="WEB" dataType="STRING"/>
  <field name="PLOT" dataType="STRING"/>
  <field name="QD" dataType="STRING"/>
  <field name="SPECIES" dataType="STRING"/>
  <field name="OBS" dataType="STRING"/>
  <field name="COVER" dataType="FLOAT"/>
</table>
</schema>
```

sqlDef:

Selected Entity: **veg_data_test.csv**

outputType: **As Field**

Target File Extension in Compressed File:

recordid: **dpennington.32.2**

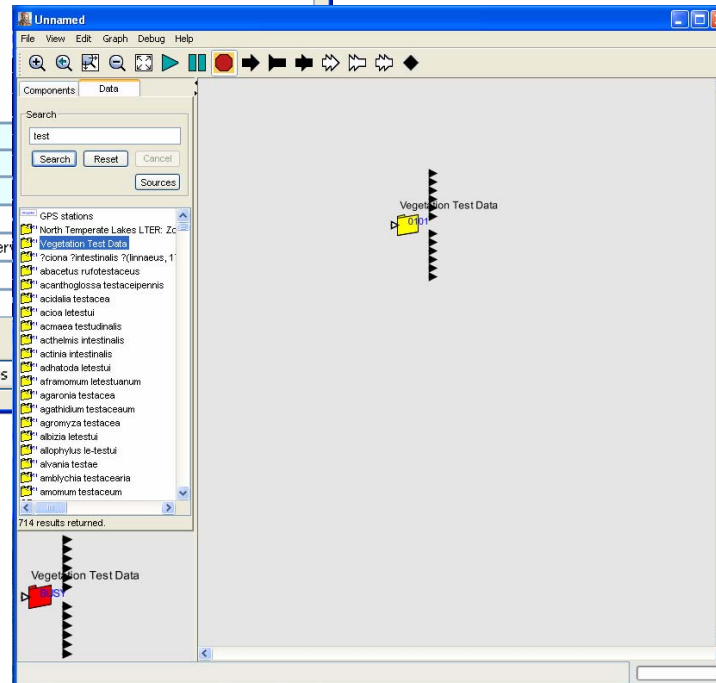
endpoint: **http://ecogrid.ecoinformatics.org/knb/serve**

namespace: **eml://ecoinformatics.org/eml-2.0.0**

veg_data_test.csv:

Commit Add Remove Restore Defaults Preferences

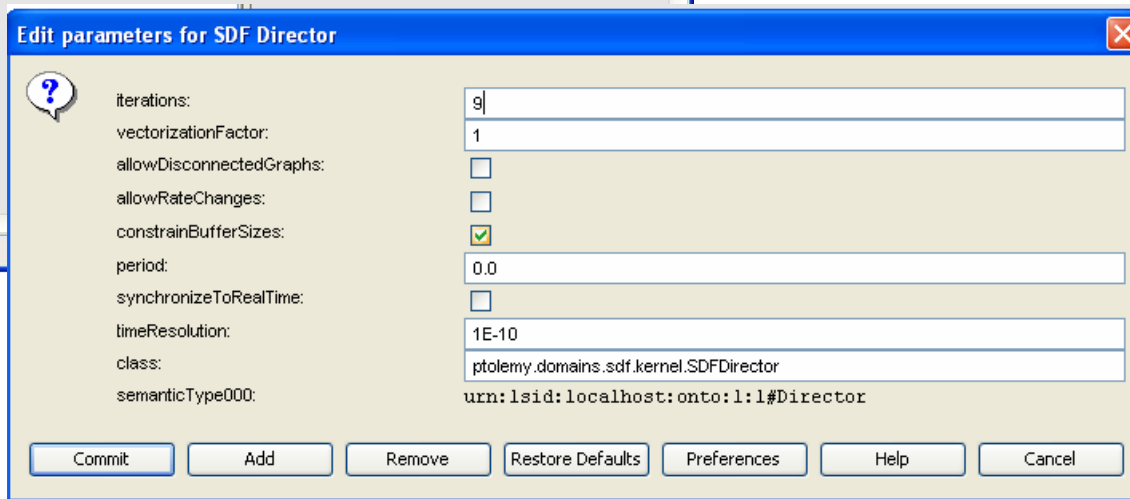
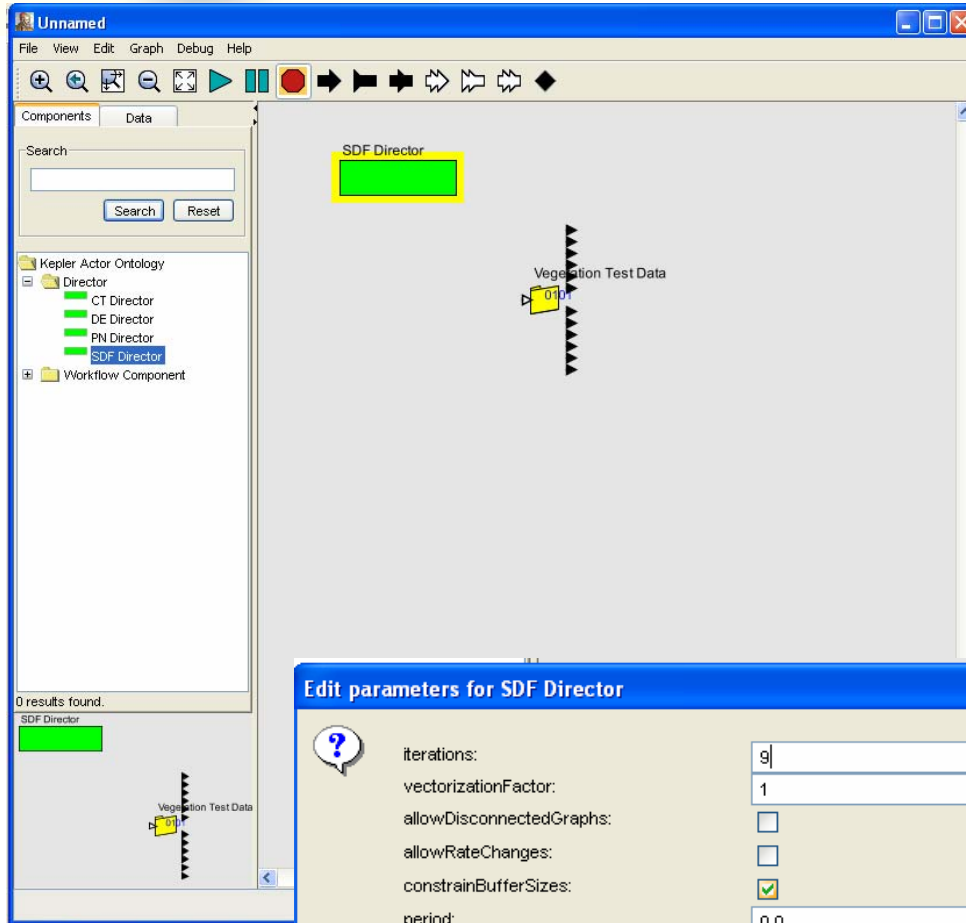
- Look at the metadata
Right-click on icon-> Get Metadata
- Show the schema
Double-click on icon or right-click->configure
- Tool tips on ports





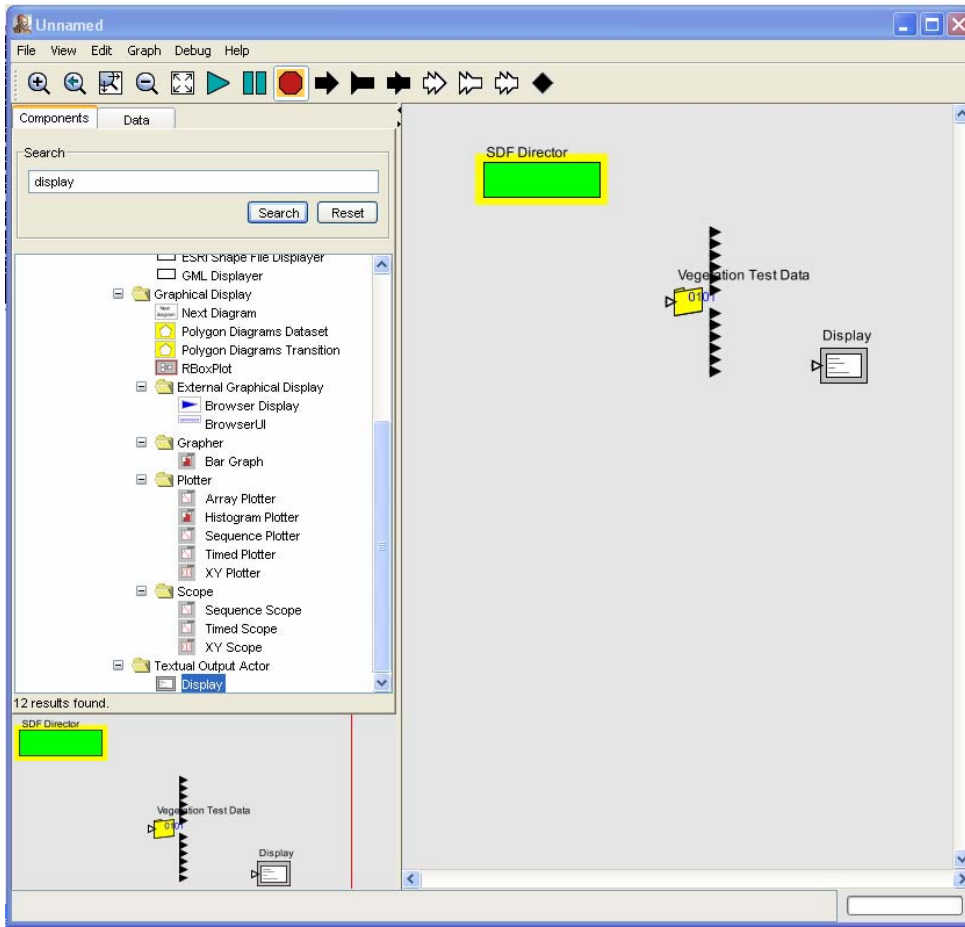
Director

- Add a director
 - Click on the Components tab
 - Double-click on Kepler Actor Ontology
 - Double-click on Directors
 - Drag and drop the SDF Director onto the canvas
- Configure director
 - Double-click or right-click->configure
 - Change iterations to 9 (the number of rows to be read in the file)
 - Click Commit to save





Display the file



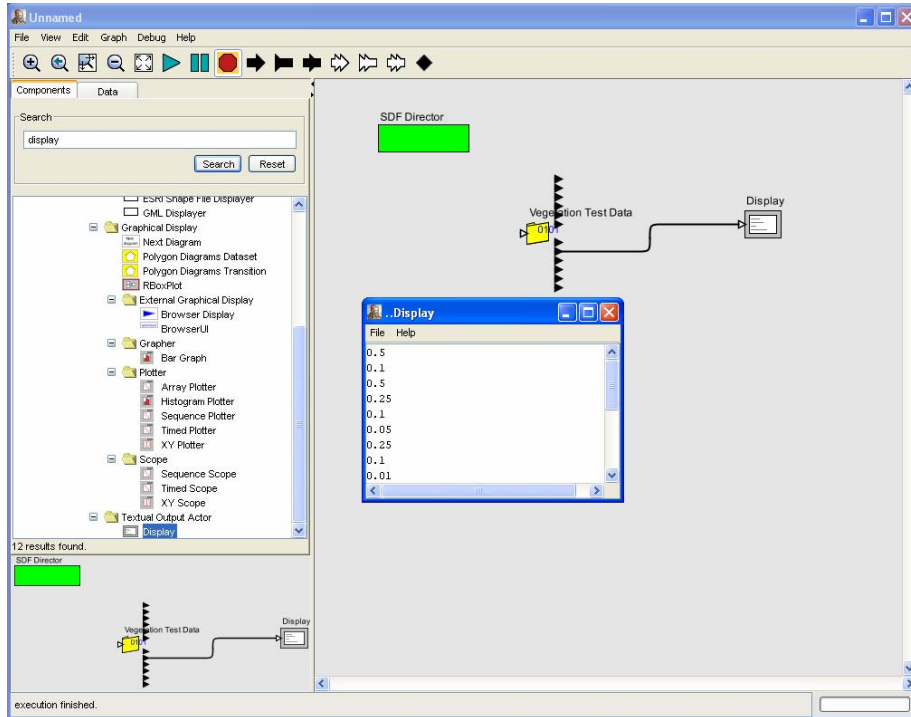
- Search for "display"
- Find the Display actor (last icon)
- Drag and drop the Display actor onto the canvas





Hooking up ports

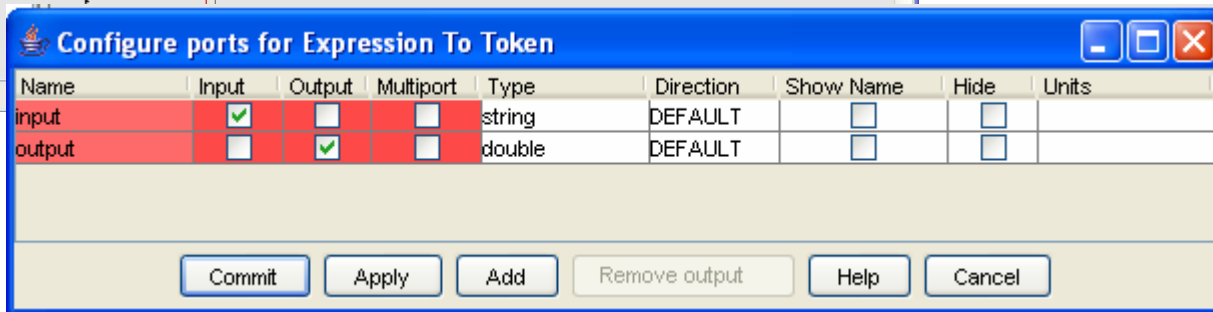
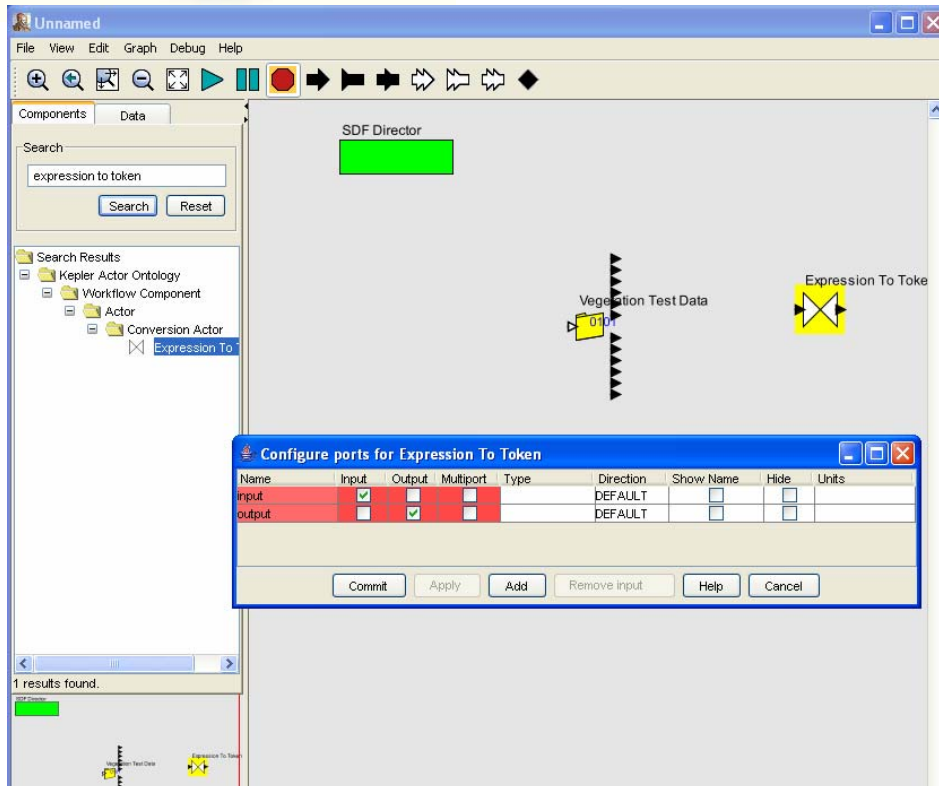
- Find the “cover” port
Use tool tips or view schema
Note string types
- Click-hold the cover port and drag to the input port on the Display actor
- Run the workflow
Click the teal triangle or
View->Run Window->Go
- Delete the Display actor





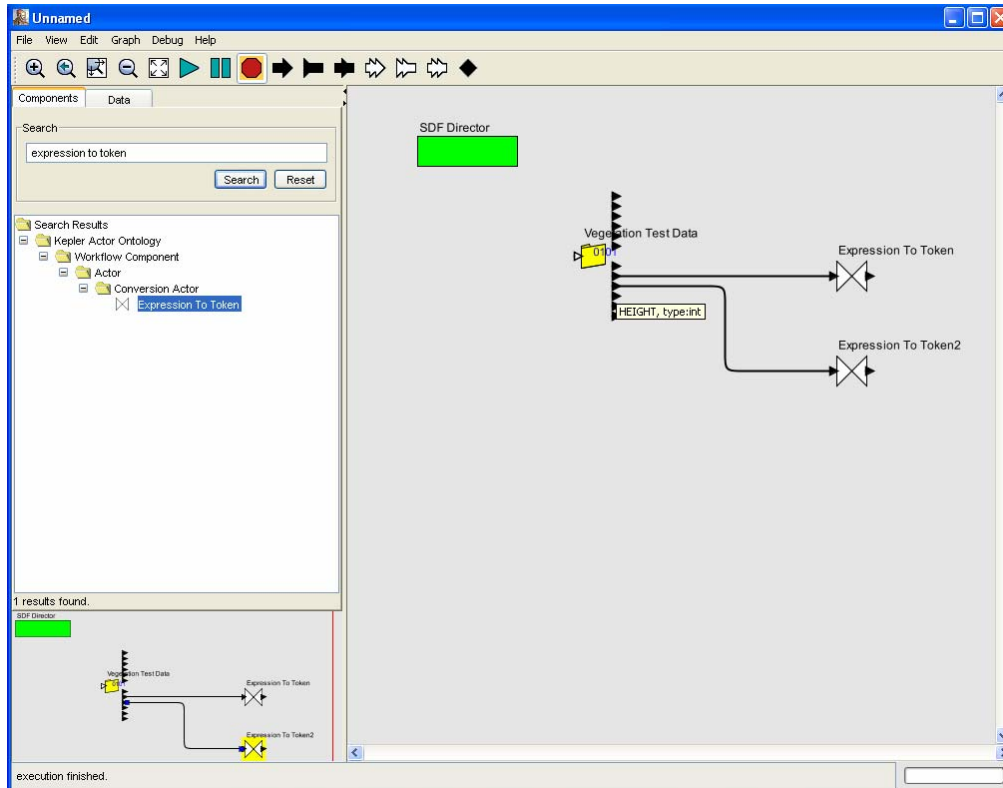
String conversion

- Search for Expression to Token actor
- Add Expression to Token actor to canvas
- Configure ports
 - Right-click->configure ports
 - Change input port Type to string
 - Change output port Type to double
 - Commit
- Hook up Cover data output port with Expression to Token input port





String conversion

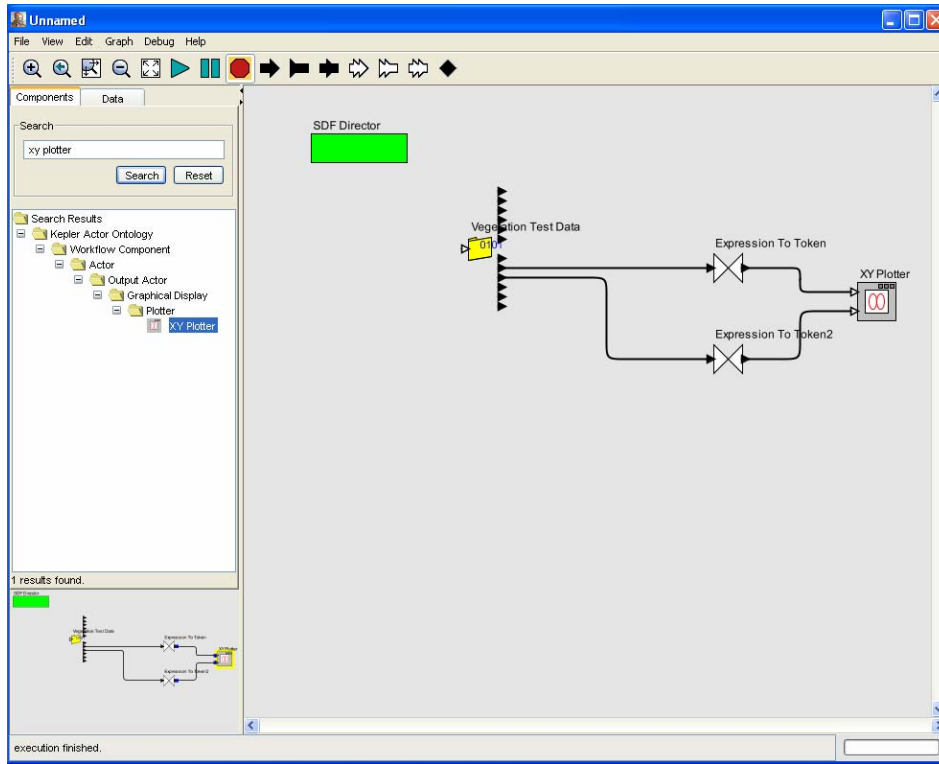


- Repeat for Height data output port
- Copy and paste Expression to Token actor (Edit-> Copy-> Paste or cntl-c cntl-v), Move copy from over original
- Hook up height data output port with Expression to Token input port





Add graph



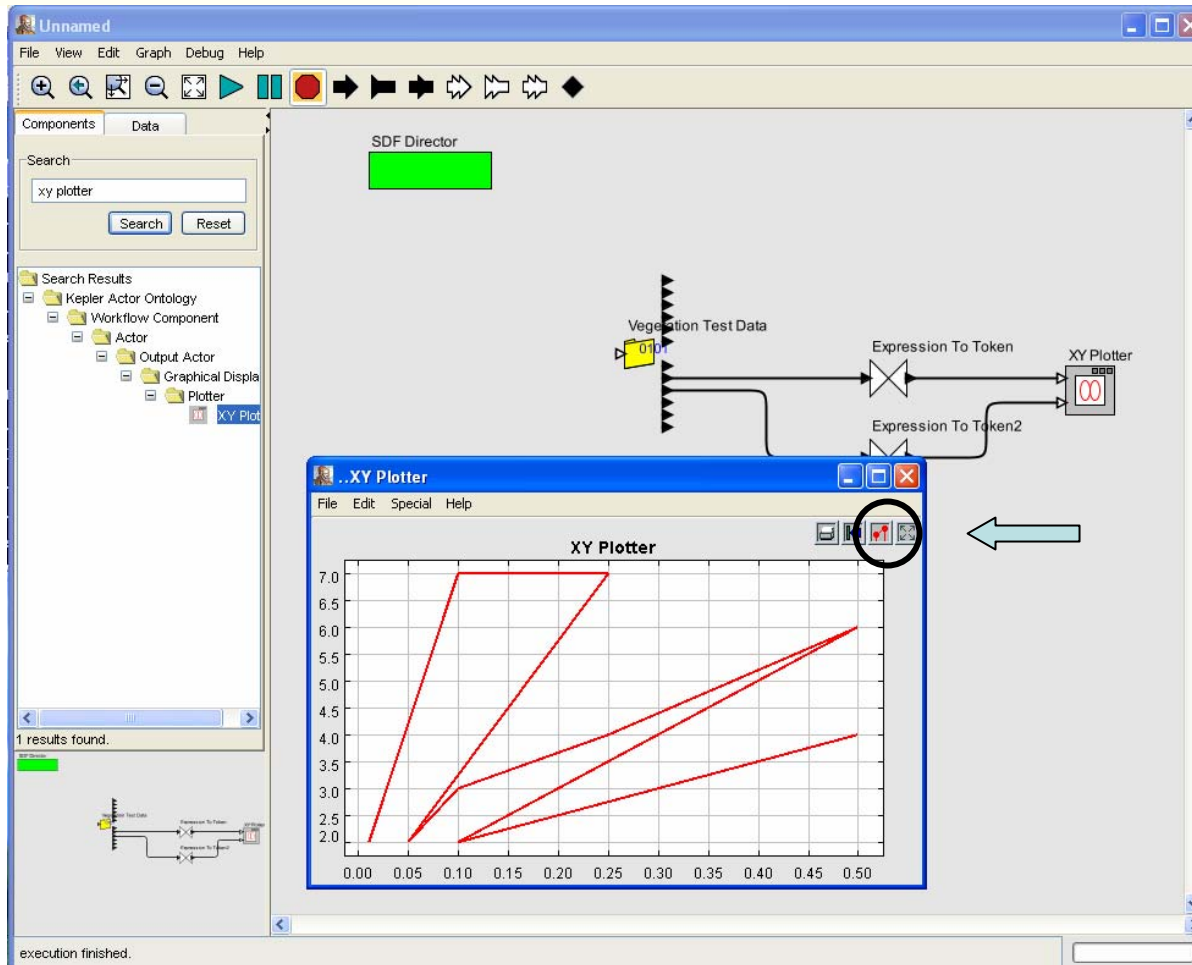
- Search for "xy plotter"
- Add XY Plotter actor to canvas
- Hook up Expression to Token output ports with XY Plotter input ports





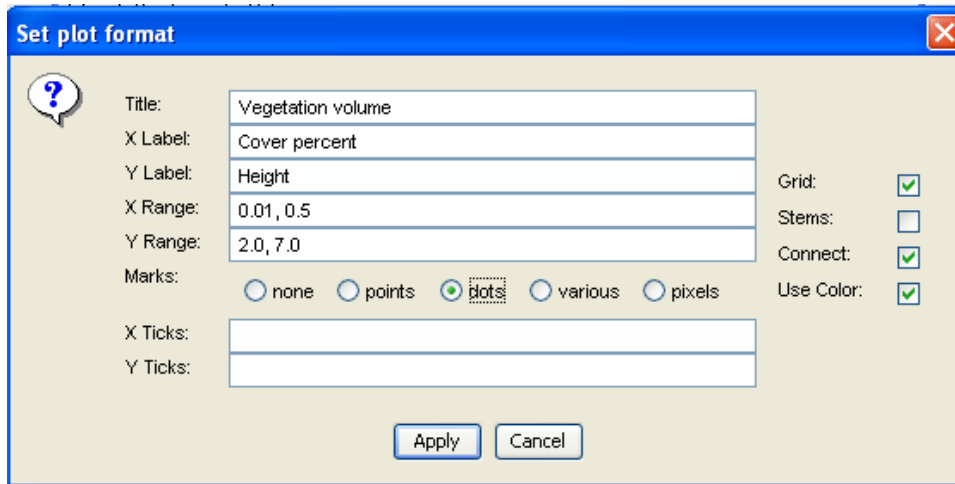
Run the workflow

- Click the Run button (red triangle)
- Note lousy plot format..close the display
- Change the plot format
View->Run Window
Click on small plot format icon





Reformat the plot



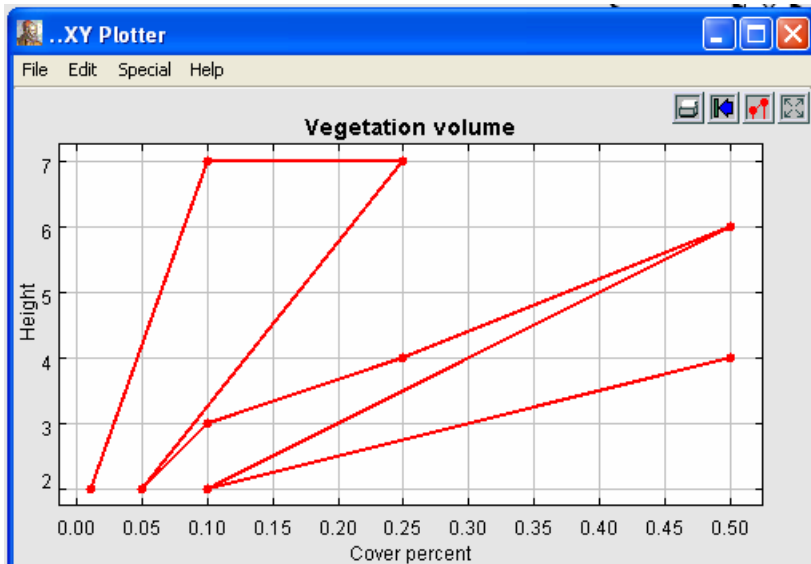
The 'Set plot format' dialog box has a blue title bar with a question mark icon. It contains several input fields and checkboxes. The 'Title' field is set to 'Vegetation volume'. The 'X Label' field is set to 'Cover percent'. The 'Y Label' field is set to 'Height'. The 'X Range' field is set to '0.01, 0.5'. The 'Y Range' field is set to '2.0, 7.0'. The 'Marks' section has radio buttons for 'none', 'points', 'dots' (selected), 'various', and 'pixels'. The 'Grid' checkbox is checked. The 'Stems' checkbox is unchecked. The 'Connect' checkbox is checked. The 'Use Color' checkbox is checked. There are 'Apply' and 'Cancel' buttons at the bottom.

Field	Value
Title	Vegetation volume
X Label	Cover percent
Y Label	Height
X Range	0.01, 0.5
Y Range	2.0, 7.0
Marks	dots
Grid	checked
Stems	unchecked
Connect	checked
Use Color	checked

- Change Title to something meaningful (I used Vegetation Volume)
- Add axis labels
- Select dots
- Unselect Connect
- Apply
- Close Run Window
- Run from run button
- Save your work

File->Save->

c:\kepler\workflows\eco
\veg_data_xxx.xml,
where xxx is your initials





Without EML

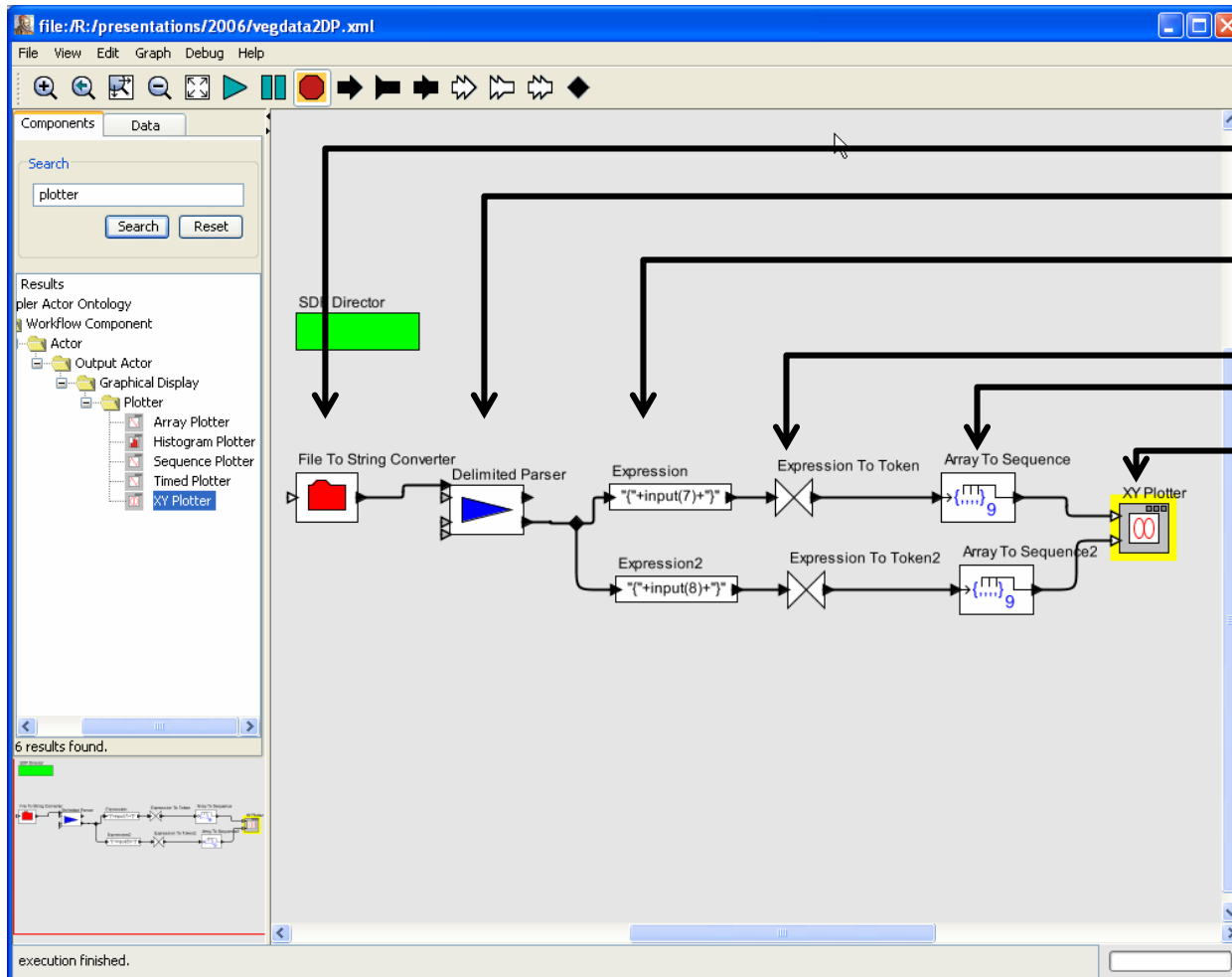
- The harder way, without EML-described metadata





Cheat-look ahead

Open vegdata2DP.xml from Kepler->File->Open File-> [locate file]

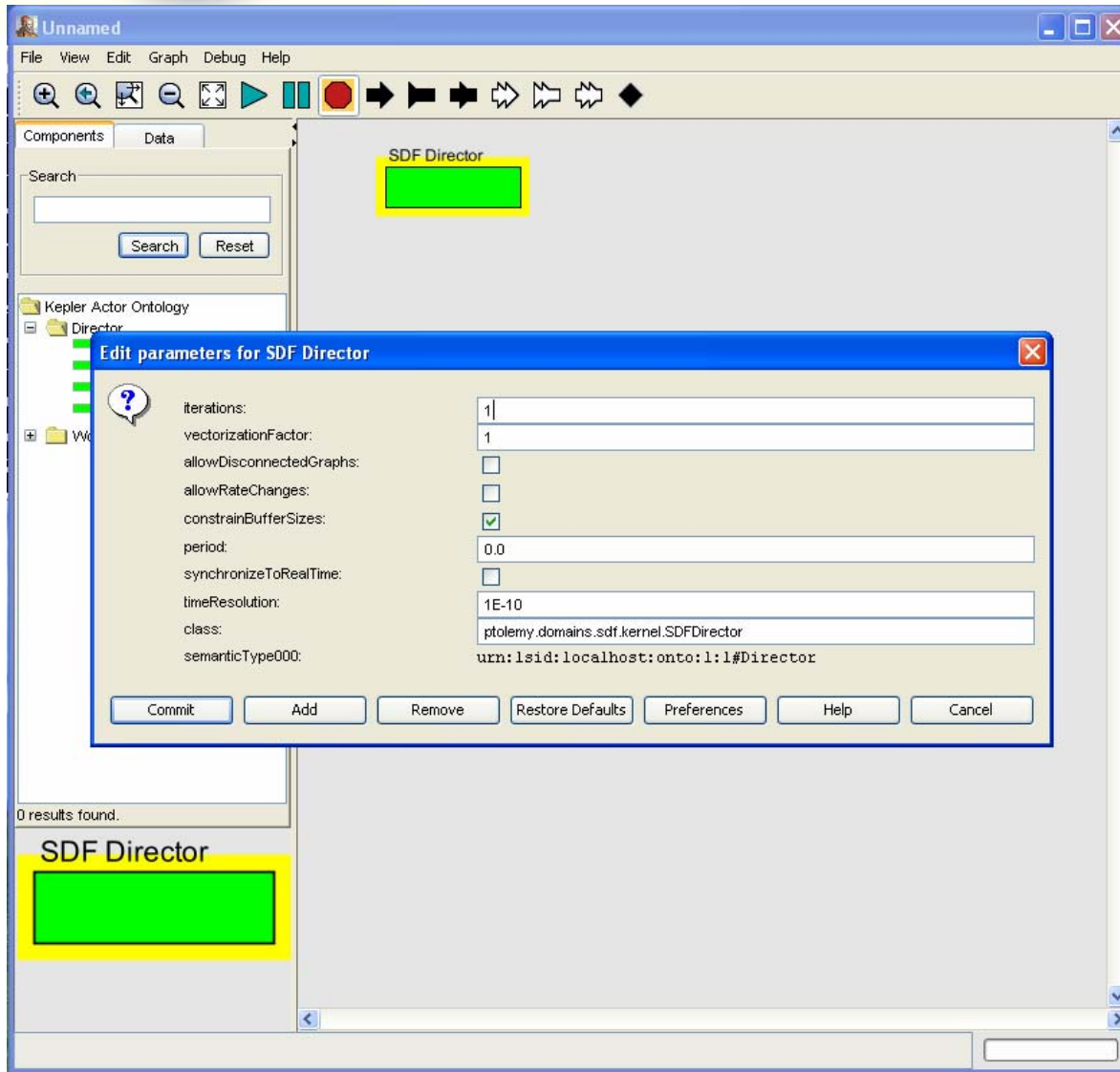


Opens the file
Separates file into columns
Extract cover and height columns
Converts to decimal number
Converts array to elements
Constructs graph
Note: all this would not be necessary if EML used





Start a new workflow



- Open the workflow editor
File->New->Graph Editor
- Select a director
Double click
Workflow
Components
Double click
Directors
Drag and drop SDF
Director
- Configure director
Right click on
director
Change number of
iterations from 0
to 1
- Commit





Read the file

- Add the "File to String Converter" actor to the canvas
- Search for "File to String"

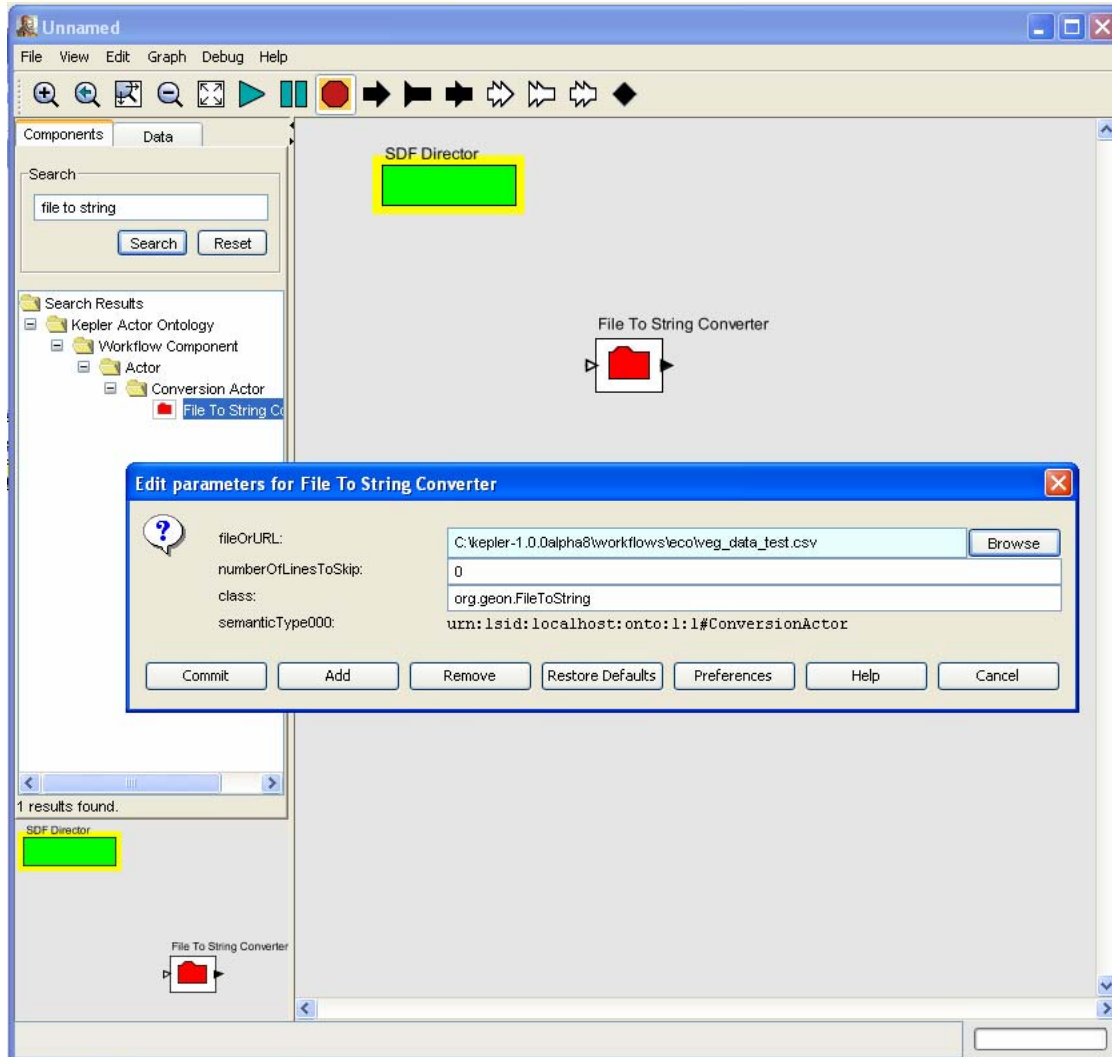
Drag and drop the File to string converter actor

- Configure the actor

Double-click or right-click->configure

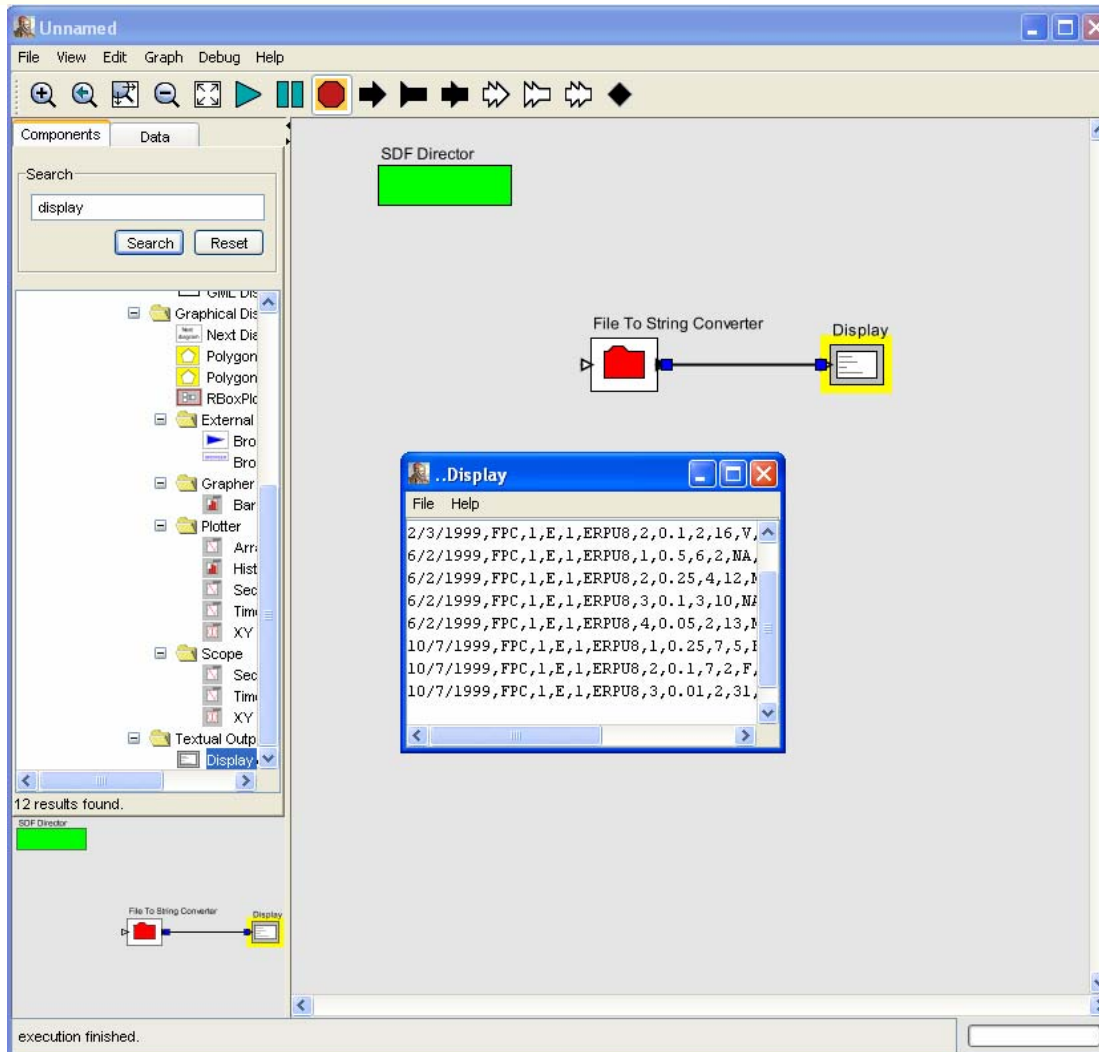
Browse for the veg_data_test.csv file

- Commit





Display to check

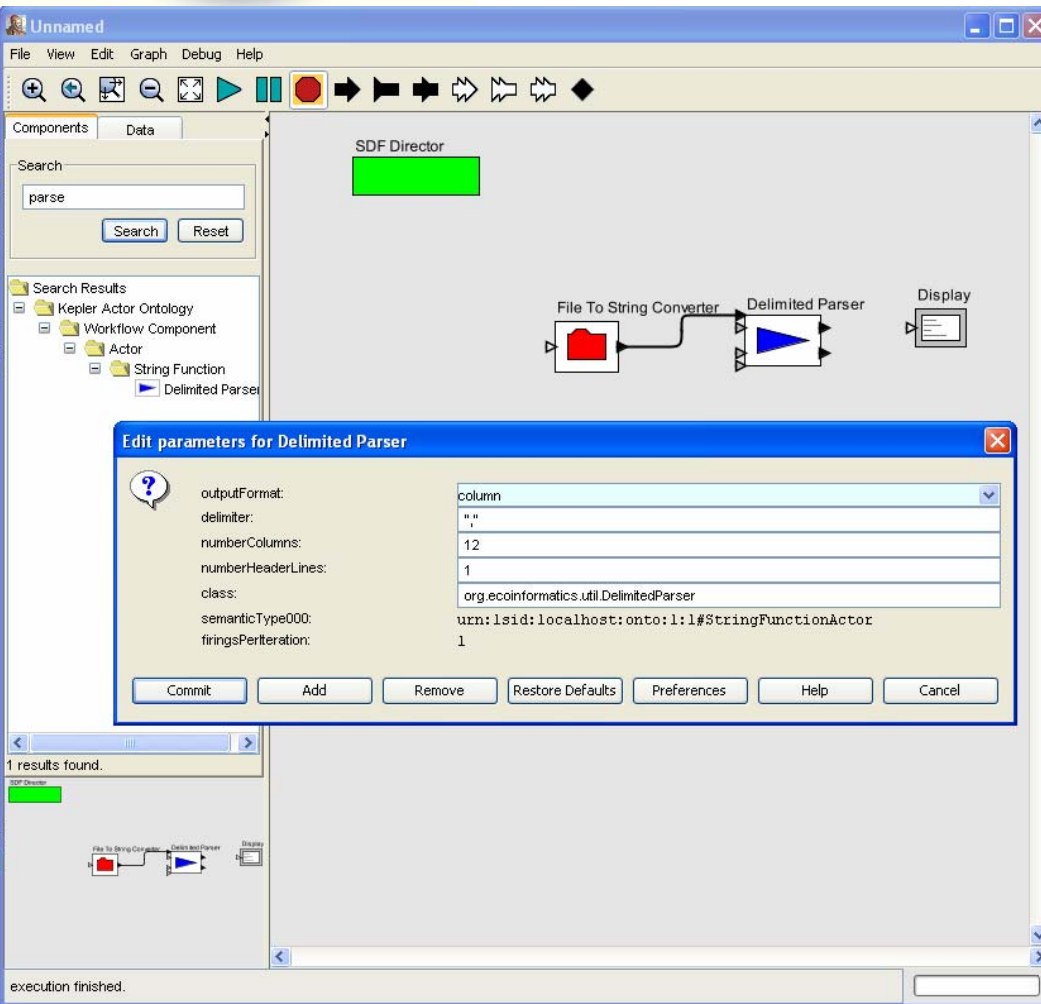


- Add a "Display actor", to check that the "file to string" actor is working
- Search for "display"
Drag and drop the Display actor
- Hook up ports between the 2 actors
Click-hold the output (right) port on the File Reader actor and drag to the input (left) port on the display actor, then release
- Run the workflow



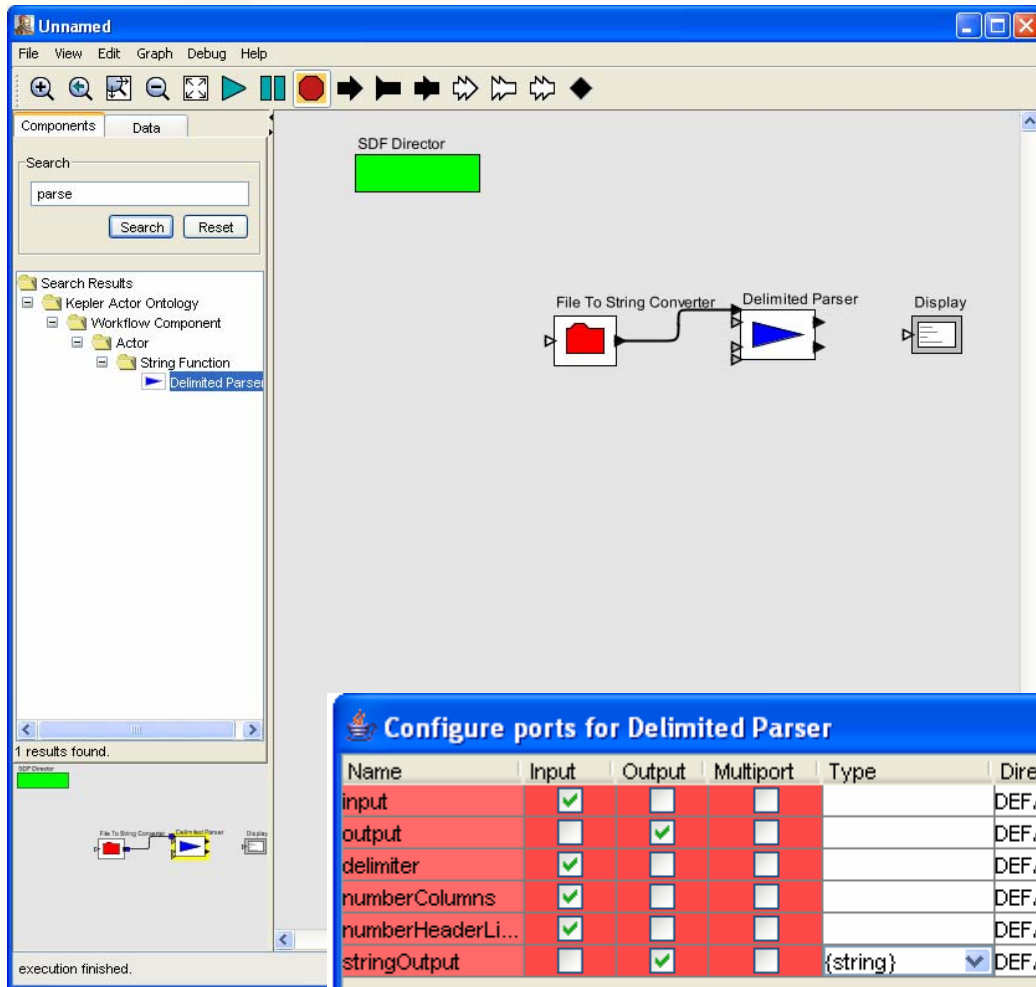
Parse the file

- Keep the Display window
- Move (but keep) Display actor
 - Delete port connection (click->delete)
 - Move actor to right
- Search for Delimited Parser actor
 - Type "parser" in text entry box
 - Click Go
- Drag and drop Delimited Parser onto canvas adjacent to File Reader
- Configure Delimited Parser
 - Double-click or right-click->Configure
 - outputFormat = column
 - Delimiter = ", "
 - numberColumns = 12 (refer to Display actor output)
 - numberHeaderLines = 1 (refer to Display actor output)
 - Commit

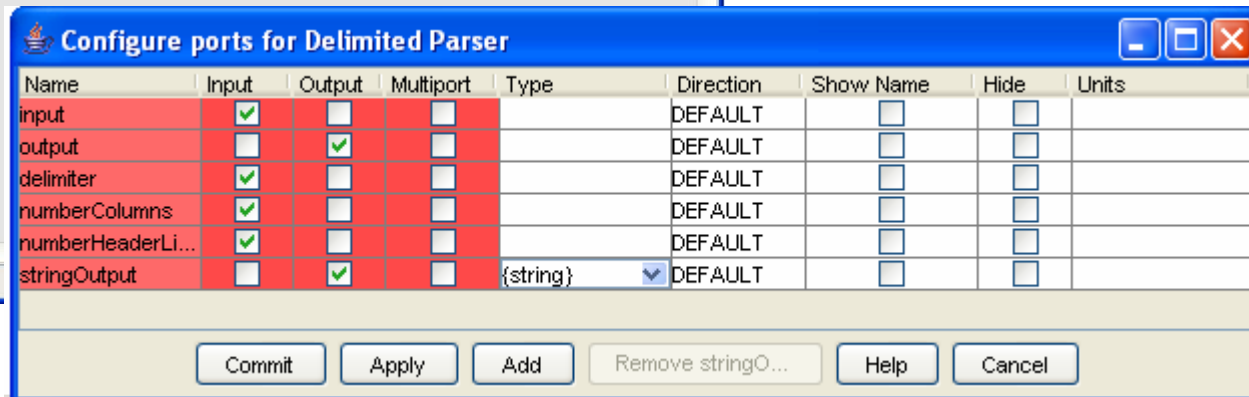




Parse the file (cont'd)

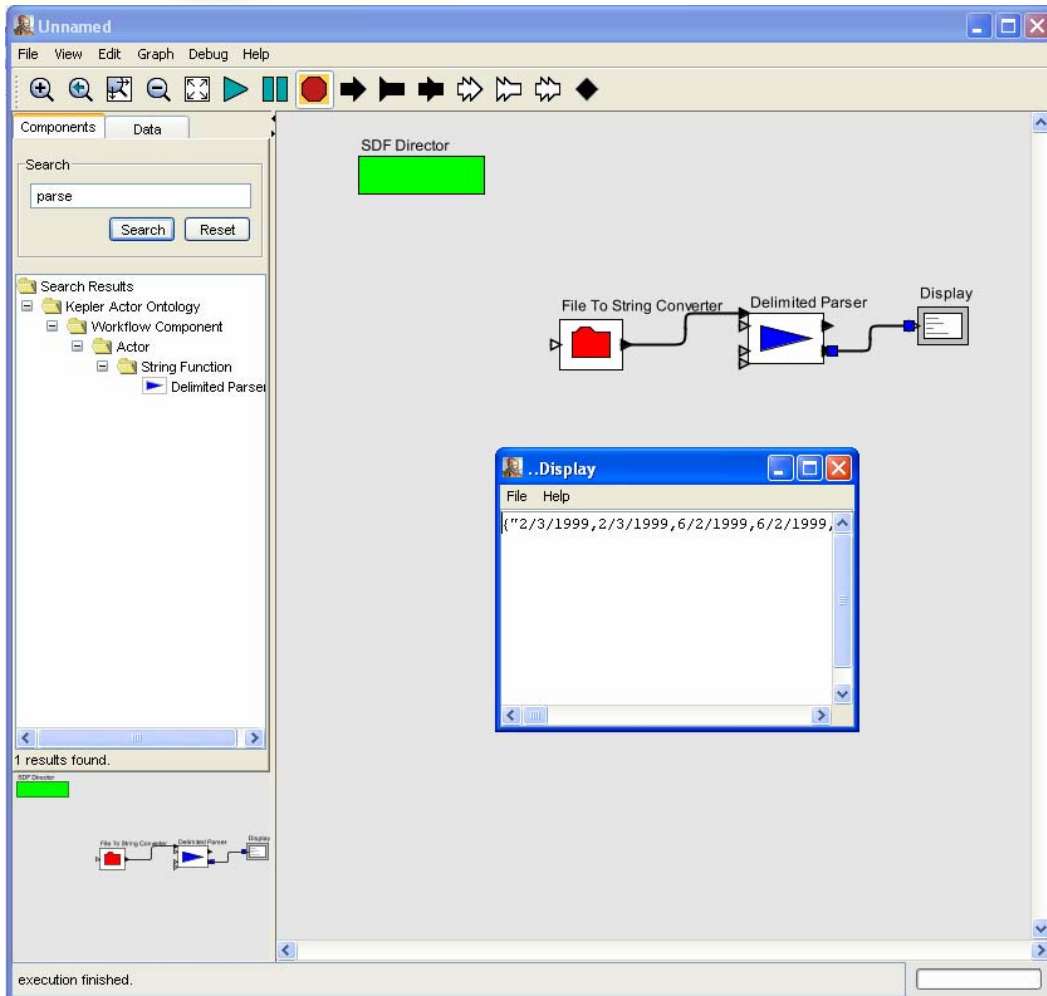


- Configure Delimited Parser output port
Right-click->configure ports
Find stringOutput output port
Change Type to {string} (braces are important..make sure to include them)
Commit
- Hook up File Reader/Delimited Parser ports
Connect File Reader output port to uppermost, black, Delimited Parser input port





Parse the file (cont'd)

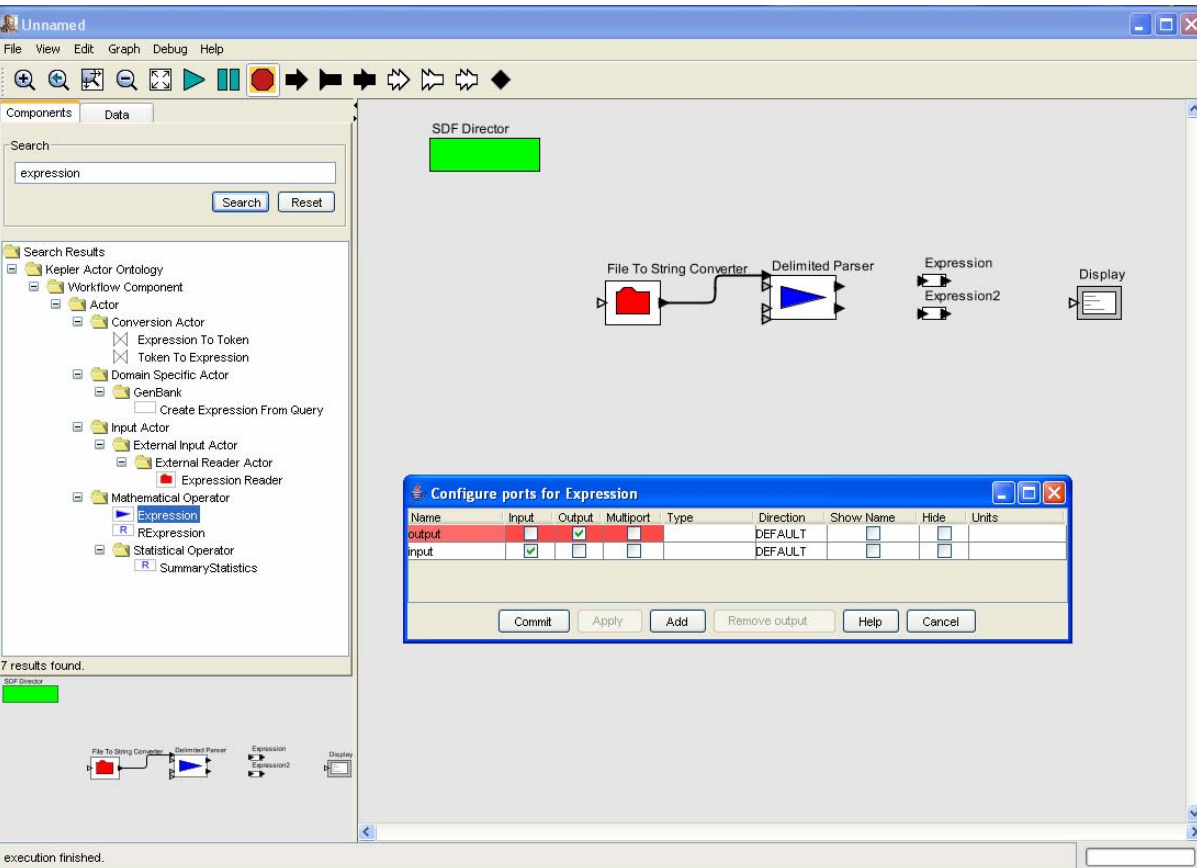


- Hook up the Delimited Parser/Display ports
Use lower, type=string output port for Delimited Parser
- Run
- Compare the display to your earlier display window
- Save your work!
File->Save->
browse to
c:\kepler\workflows\eco\
name as veg_data2_xxx.xml,
where xxx is your initials





Extract columns

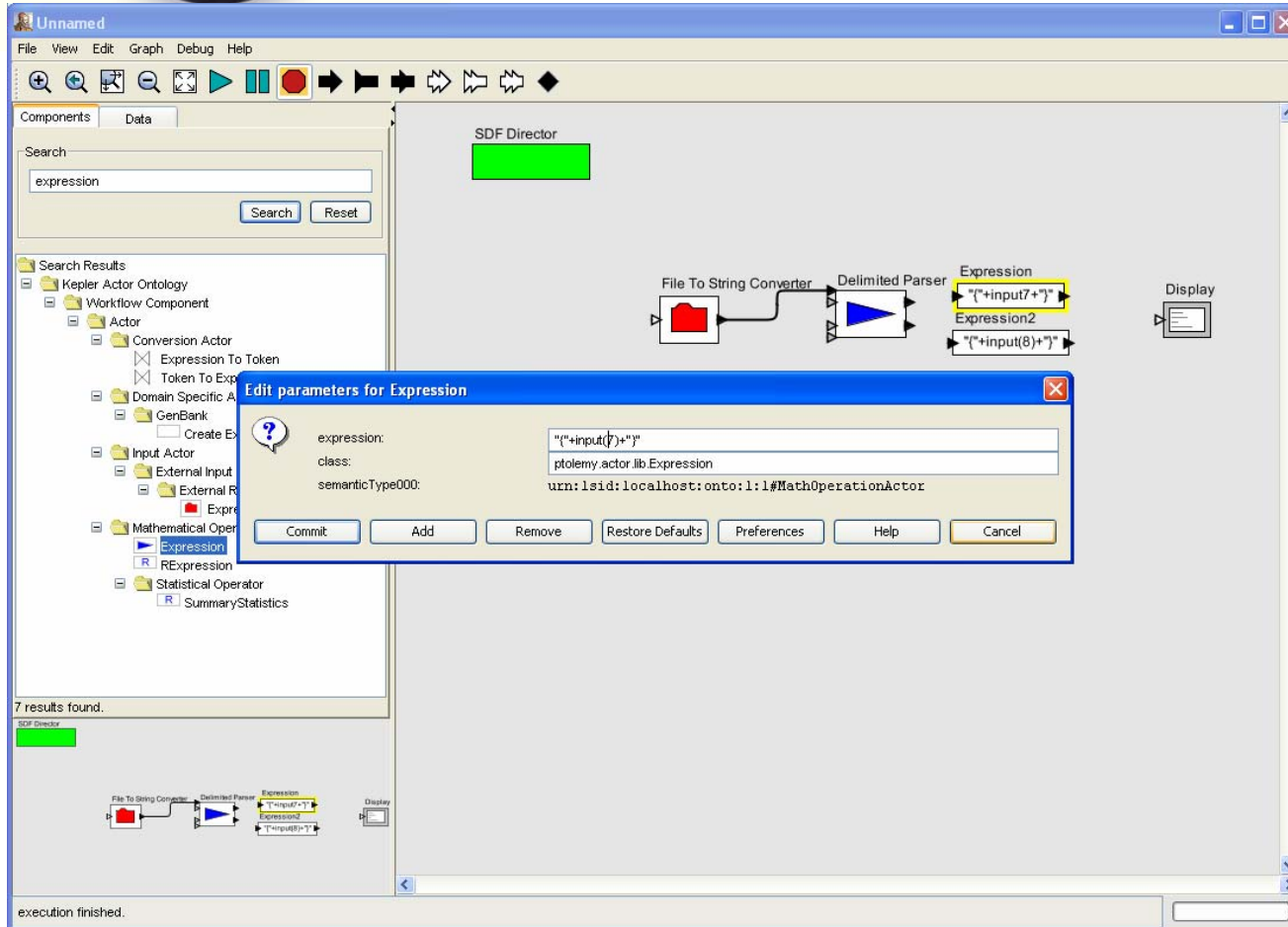


- Delete the connection to the Display actor
- Move (but keep) Display actor
- Locate Expression actor
Search "expression" or
Actors->Mathematical
Operations
- Add Expression actor to
canvas two times
- Add input port to both
Expression actors
Right-click->Configure
ports->Add
Check Input box for new
port
Click blank box under
Name
Type port name "input"
Commit





Extract columns-cont'd

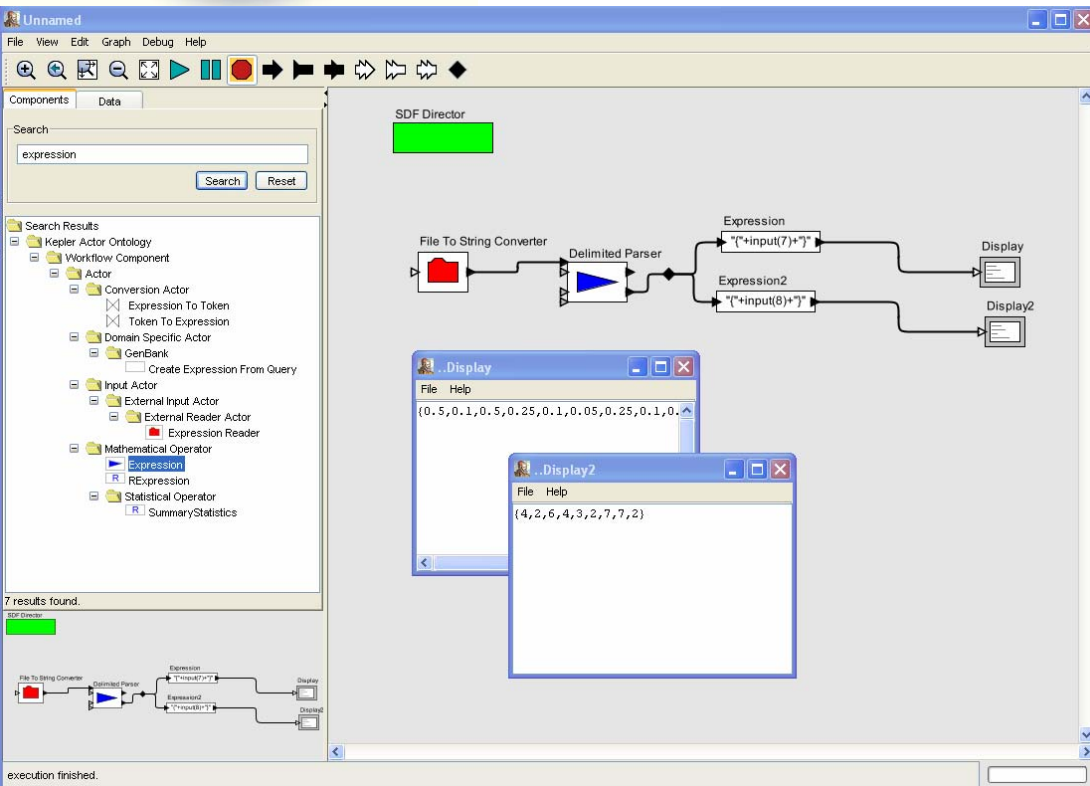


- Configure Expression actors
- Double-click or right-click->configure
- Type in first actor:
`"{" +input(7)+ "}"`
- Type in second actor:
`"{" +input(8)+ "}"`
- Commit





Extract columns-cont'd



- Add relation
Click (one time) on diamond on task bar
Hook Delimited Parser lower (stringOutput) output port to relation
- Hook relation to Expression input ports
Hint: it's easier to click on the Expression port and drag to the relation than vice versa
- Copy the Display actor
Click on it
Copy and paste (Edit menu or shortcuts)
Move copy so both show
- Hook output ports on each Expression actor to the input port of a Display actor
- Run
- Move the top Display window so you can see both





Convert to decimal number

The screenshot shows the SEEK software interface. The main window displays a workflow diagram with components: File To String Converter, Delimited Parser, Expression (with formula $input(7)+input(8)$), Expression2 (with formula $input(8)+input(7)$), Expression To Token, Expression To Token2, Display, and Display2. A search bar on the left contains the text "expression to token". Below it, the search results show "Expression To Token" under the "Conversion Actor" category. Two small windows are open: ".veg_data_ddp.Display2" showing the array $\{4, 2, 6, 4, 3, 2, 7, 7, 2\}$ and ".veg_data_ddp.Display" showing the array $\{0.5, 0.1, 0.5, 0.25, 0.1, 0.05, 0.25, 0\}$. The status bar at the bottom indicates "execution finished."

- Add Expression To Token actor two times
Search for "token"
- Configure ports
Input Type=string
Output Type={double}
Commit
- Hook up Expression output ports to Expression To Token input ports
- Hook up Expression To Token output ports to Display input ports
- Run

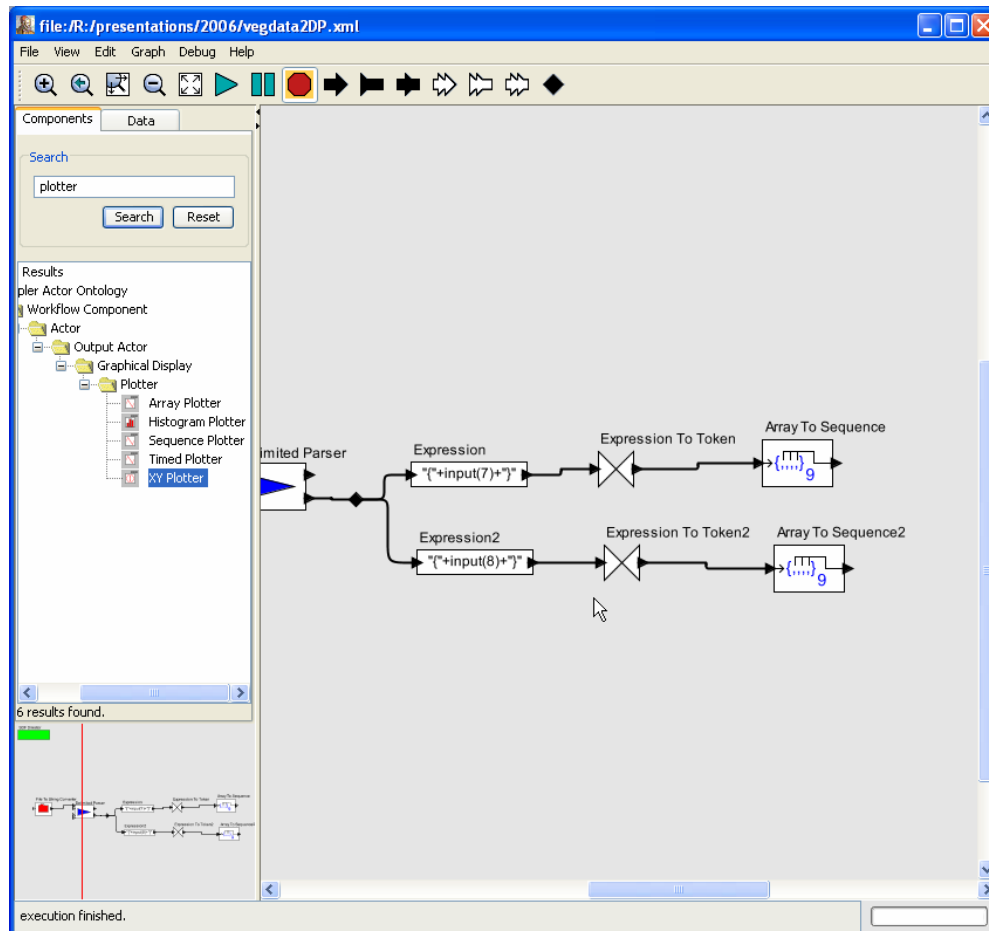
The screenshot shows the "Configure ports for Expression To Token2" dialog box. It contains a table with columns: Name, Input, Output, Multipoint, Type, Direction, Show Name, Hide, and Units. The table has two rows: "input" and "output". The "input" row has a checked box in the Input column and a type of "string". The "output" row has a checked box in the Output column and a type of "{double}". Below the table are buttons for Commit, Apply, Add, Remove output, Help, and Cancel.

Name	Input	Output	Multipoint	Type	Direction	Show Name	Hide	Units
input	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	string	DEFAULT	<input type="checkbox"/>	<input type="checkbox"/>	
output	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	{double}	DEFAULT	<input type="checkbox"/>	<input type="checkbox"/>	





Convert from array to sequence (required by xy plotter)



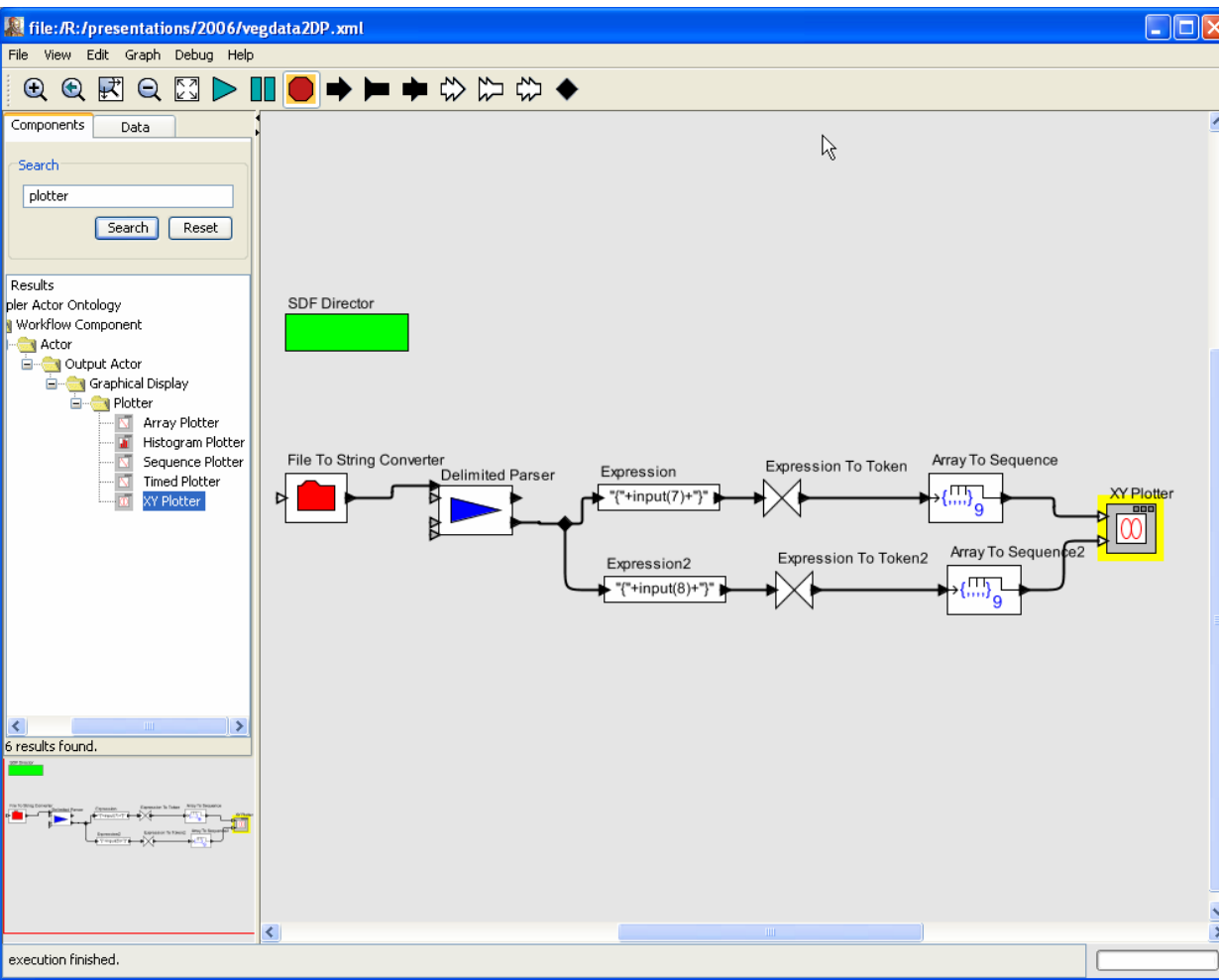
- Search for Array To Sequence actor
- Add twice to canvas
- Link Expression To Token output ports to Array To Sequence input ports
- Configure array length
Double-click, change `arrayLength` to 9
- Add Display actors (optional)
- Run – ensure there are no errors





Add XY Plotter

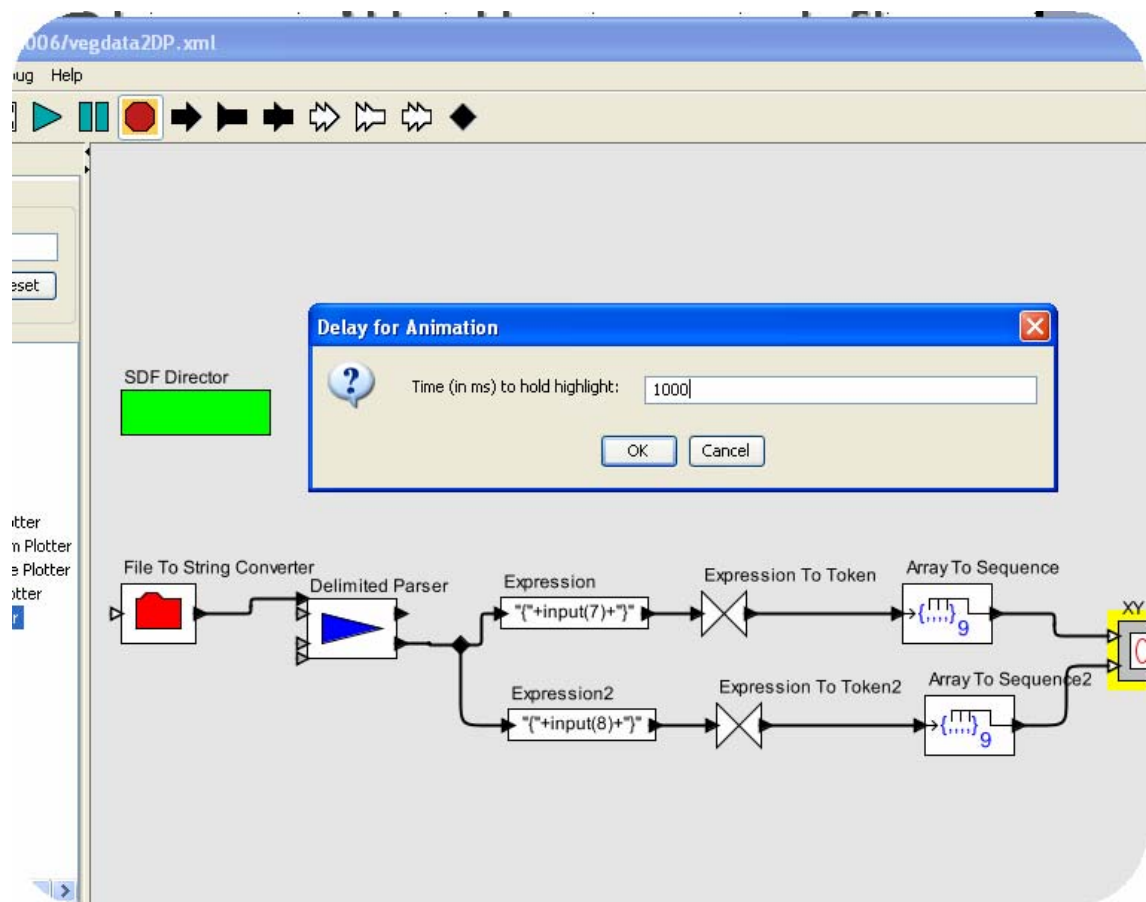
- Search for XY Plotter actor
- Add to canvas
- Connect Array to Element output ports to XY Plotter input ports
- Run
- Edit Plot format
 - From View-> Run Window





Play with the workflow

- Run workflow (click the teal triangle)
- Move the top graph so both can be viewed
- Animate the execution
Debug->Animate
execution – set for 1000 ms
- Run workflow...move graph windows so you can see the animation
- Close the graph window
- Run from run window - View->Run Window->Go





Conclusion

- Take home message....

Use EML to describe datasets...
Some pain now vs. much pain later

