What is the graph widget?

- Allows visual editing of instances and relationships between instances
- Alternative to Protege “Forms” for entering instance data
When is the graph widget appropriate?

- When instances of a slot are connected as values of some slots (e.g. a linked list where one instance is linked to another through a slot relation)

- When instances of a slot are related by instances of the `:DIRECTED-BINARY-RELATION` system class
When is the graph widget appropriate?

- Speeds knowledge acquisition in ontologies with heavily interconnected concepts
- Helps convey meaning and organization of acquired knowledge
- Data that resembles process diagrams, flow charts, organizational charts
Open the “GraphWidgetExample” project

Select the Organization class
Create the "organization_chart" slot
Configure organization_chart slot to use graph widget
How do I optimize size for the graph widget?

- Double-click on the Form background (anywhere without a widget)
- Use the Layout tab to fill horizontal and vertical space
How do I configure node appearance?

- Double-click on the organization_chart slot to bring up the widget configuration dialog.
How do I configure node appearance?

- Use the “General” tab to change the label or tooltip for the “organization_chart” slot.

- Use the “Nodes” tab to assign shapes, colors, and text properties to nodes.
Configure simple connectors

- Graph widget has two connector types, one of which is a “simple connector”

- Simple connectors have no underlying instances
Select the Editor class

Create the “responsible_for” slot (type Instance, cardinality multiple)
Configure simple connectors

- Choose connector slot for Editor class in widget configuration dialog
Drag and drop nodes to create instances of employees – draw connectors to fill in slot values.
Graph widget UI tips

- Click on node labels to drag nodes
- Nodes are resizable
- Double-click node labels to rename
- Right-click connectors to insert points
- Automatic layout provided
- Save graph as image
Graph widget UI tips

- The graph widget only allows you to draw *valid* connectors between nodes.

- You *can* add pre-existing instances to the graph widget.
Configure reified relations

- Reified relations are the second connector type offered by the graph widget

- Allows storage of additional information about a relationship between two nodes
Subclass :DIRECTED-BINARY-RELATION to use reified relations in the graph widget.
Designate a slot to hold instances of reified relations
Use the reified relations tab to configure the relation slot, line types, colors, arrowheads, etc.
Reified relation between two instances of the Employee class.
Graph widget UI Tips

- Double-click on complex links to bring up instance forms

- Designate display slots for subclasses of `:DIRECTED-BINARY-RELATION` to enable editing of connector labels
Multiple types of complex connectors are possible.
OntoViz Tab

- Tab plug-ins provide larger scale visualization of ontologies (OntoViz, TGViz, Jambalaya, etc.)

- OntoViz uses Graphviz from AT&T Research ([http://www.graphviz.org](http://www.graphviz.org))

OntoViz Tab: (helpful key for tab abbreviations)

- sub = subclass closure
- sup = superclass closure
- slx = slot extension
- isx = inverse slot extension
- slt = slots
- sle = slot edges
- ins = instances
- sys = system frames
OntoViz Tab

- Use the "add class" button to choose a class to graph.
- Use the "Options" button to configure graph.
Click the “Create Graph” button to generate graph.
TGViz Tab

- Utilizes TouchGraph (renders networks as interactive graphs)

- TouchGraph uses “Spring Layout”

- PubMed uses TouchGraph to visualize graphs of related documents in medical libraries
Use the “add class” and “add instance” buttons to add items – click the “create graph” button to graph.