

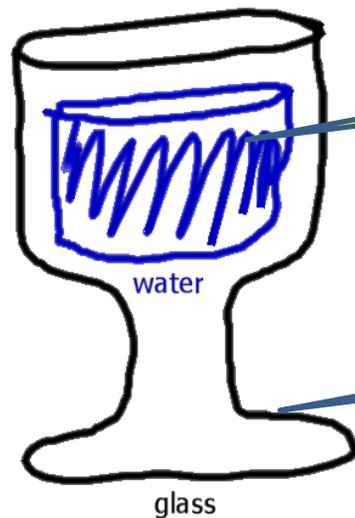
# CogSketch Basics

# This Section

- What's in a sketch?
- Starting a sketch
- Drawing glyphs
  - Inking
  - Conceptual labeling
- Layers
- Subsketches & the metalayer

# Sketches are made of Glyphs

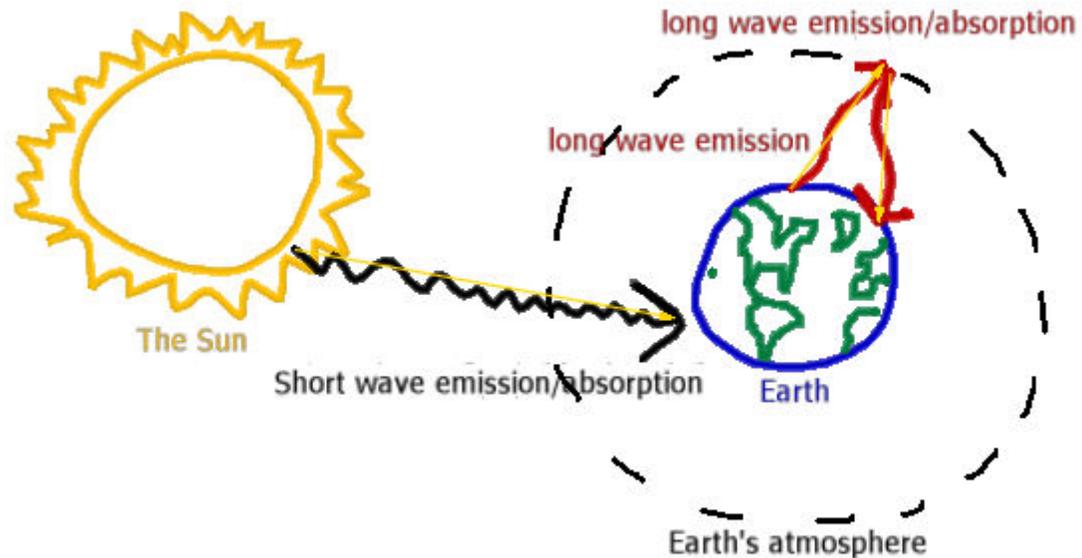
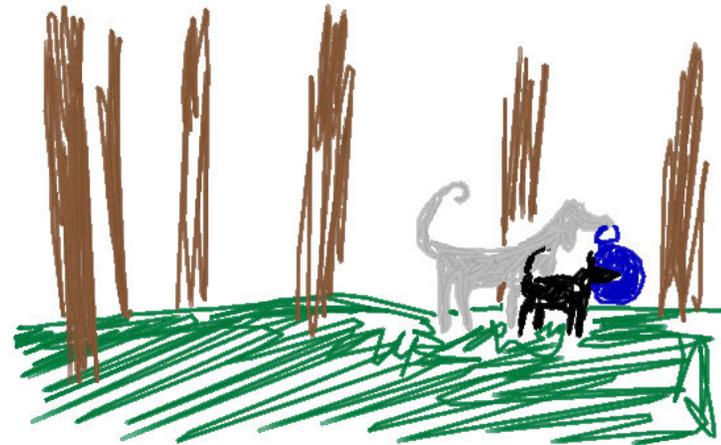
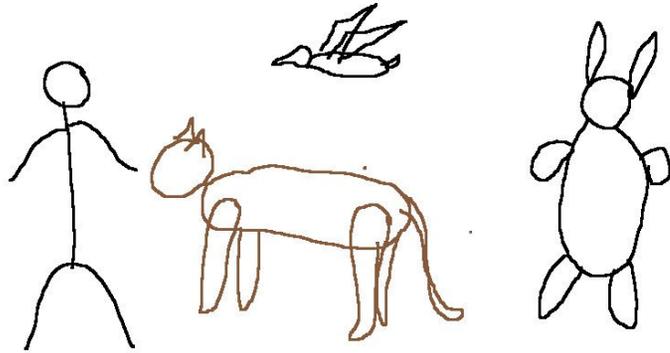
- A glyph has
  - Ink: Colored polylines
  - Content: A token representing what is depicted by the ink

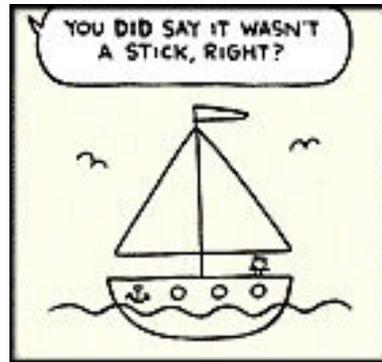
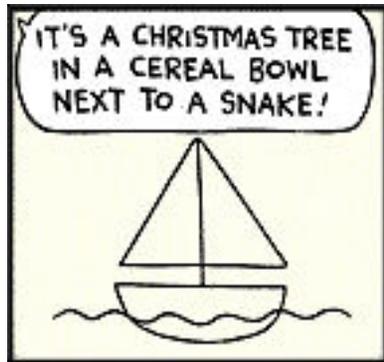
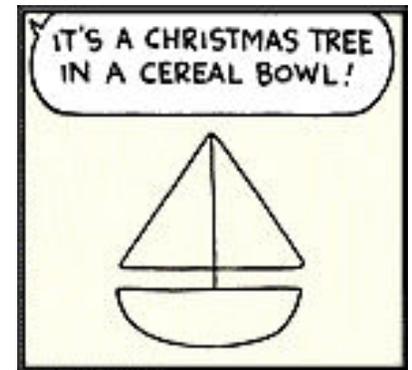
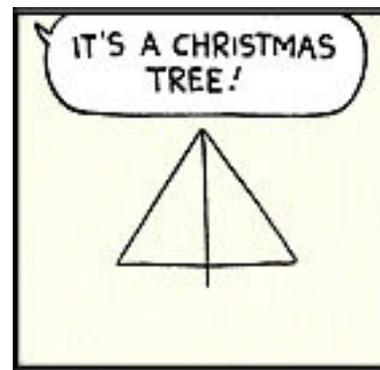
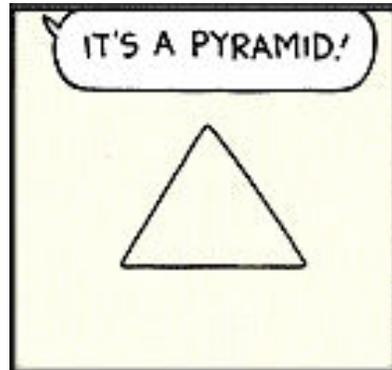
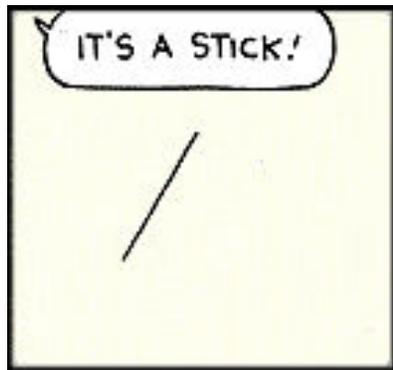


```
(nameString Object-12 "water")  
(isa Object-12 Water)  
...
```

```
(nameString Object-11 "glass")  
(isa Object-11 GlassStemware)  
...
```

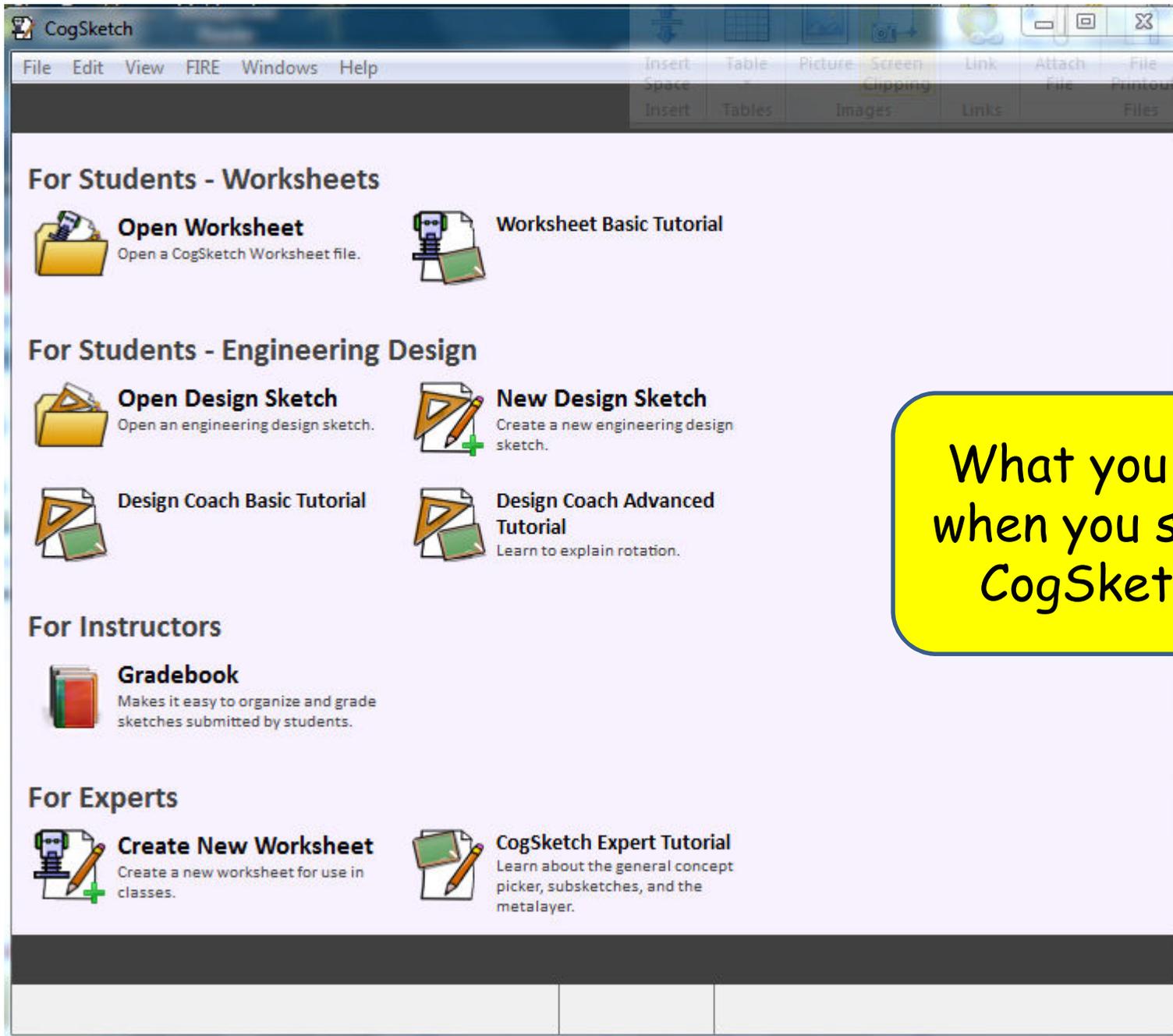
# Examples of Glyphs





# Sketches have Structure

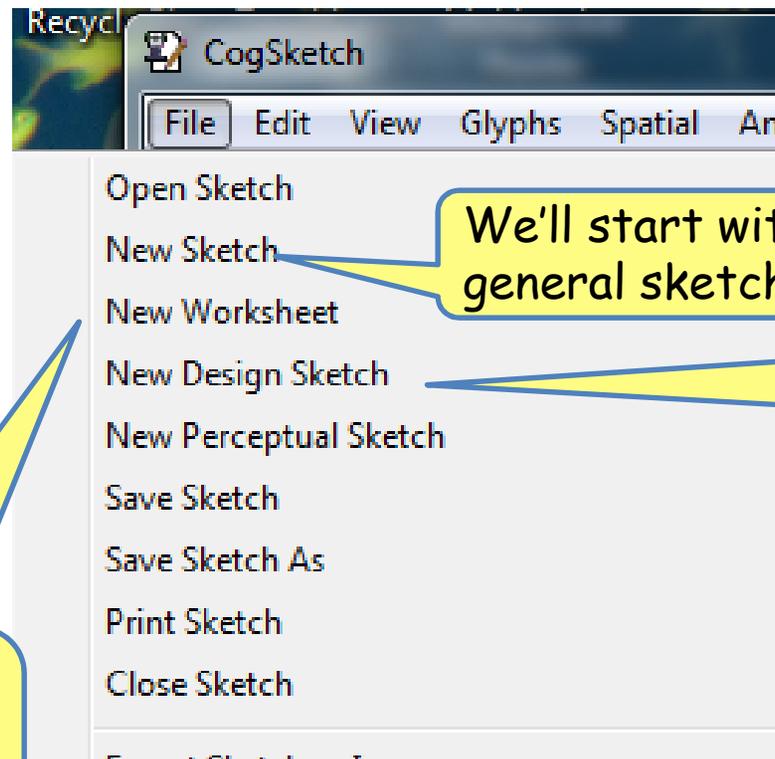
- People often draw several closely related sketches
  - Different perspectives on the same situation
  - A sequence of behaviors
  - Alternative solutions to be compared
- CogSketch captures this via *subsketches*
  - A sketch consists of one or more subsketches
  - A visual language is provided for relating them



What you see  
when you start  
CogSketch

# Creating a New Sketch

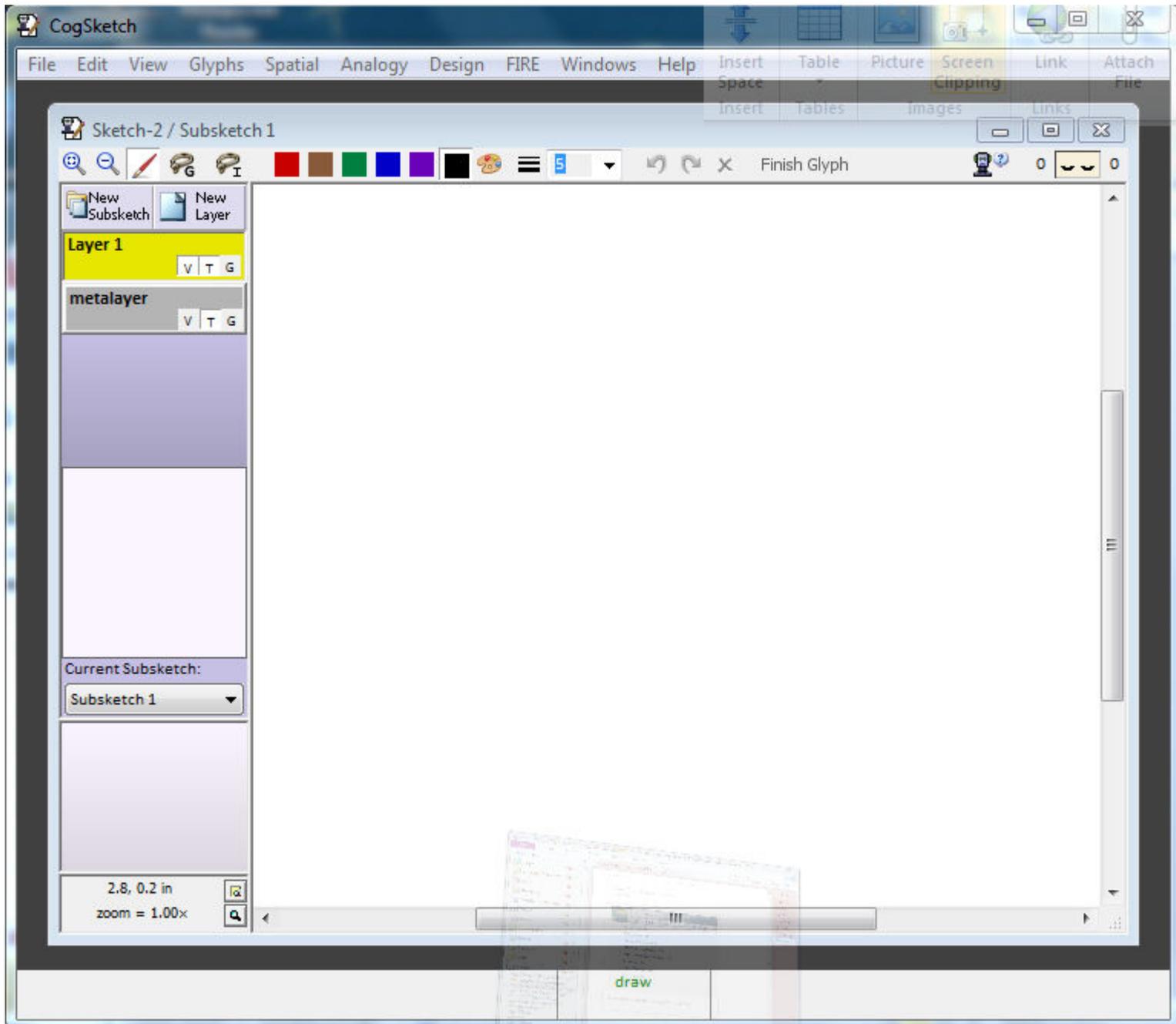
- There are four types of sketches



We'll start with  
general sketches

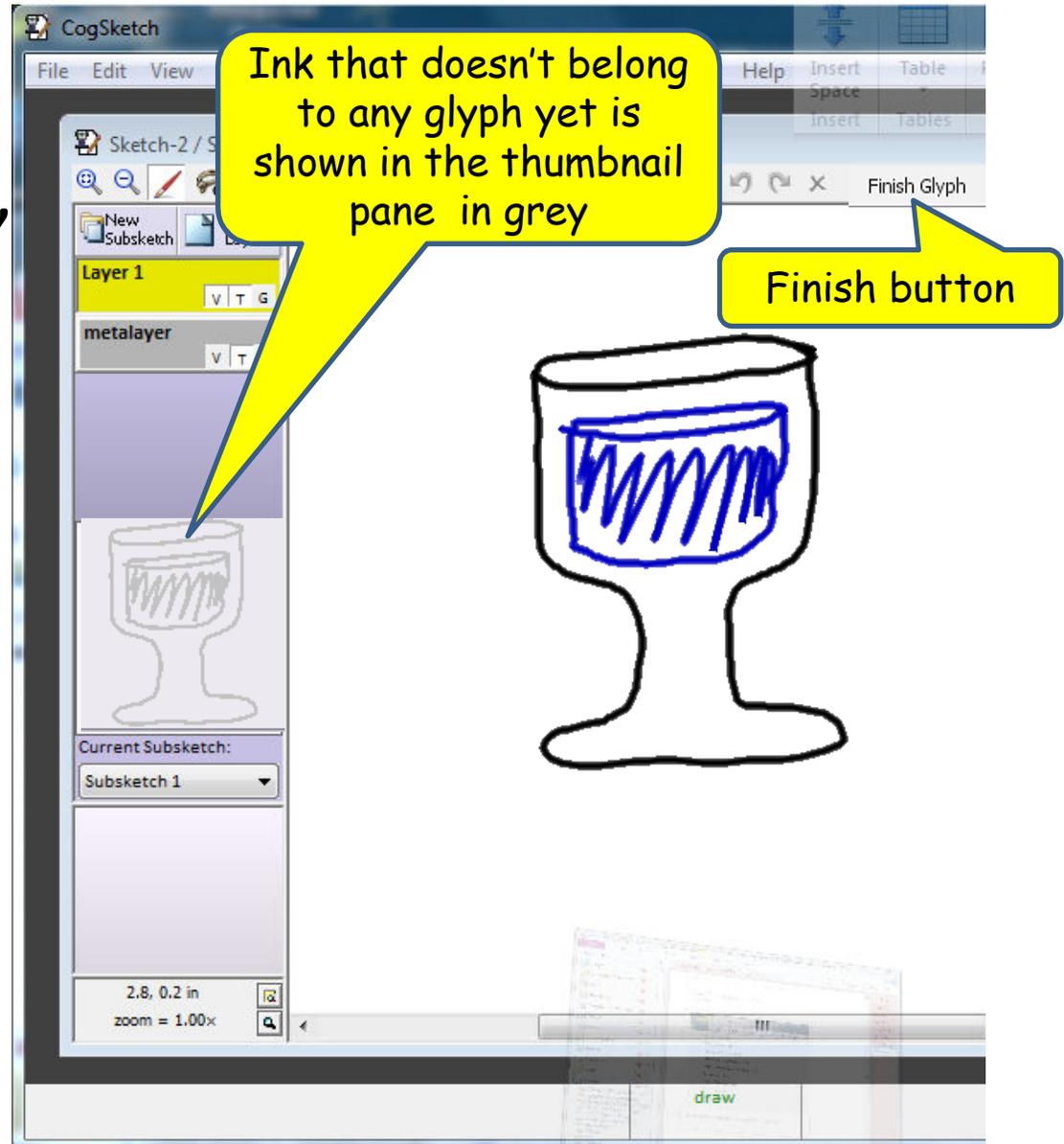
Perceptual  
Sketchpad and  
Design Coach will be  
discussed later

Worksheet  
authoring will  
be discussed  
later



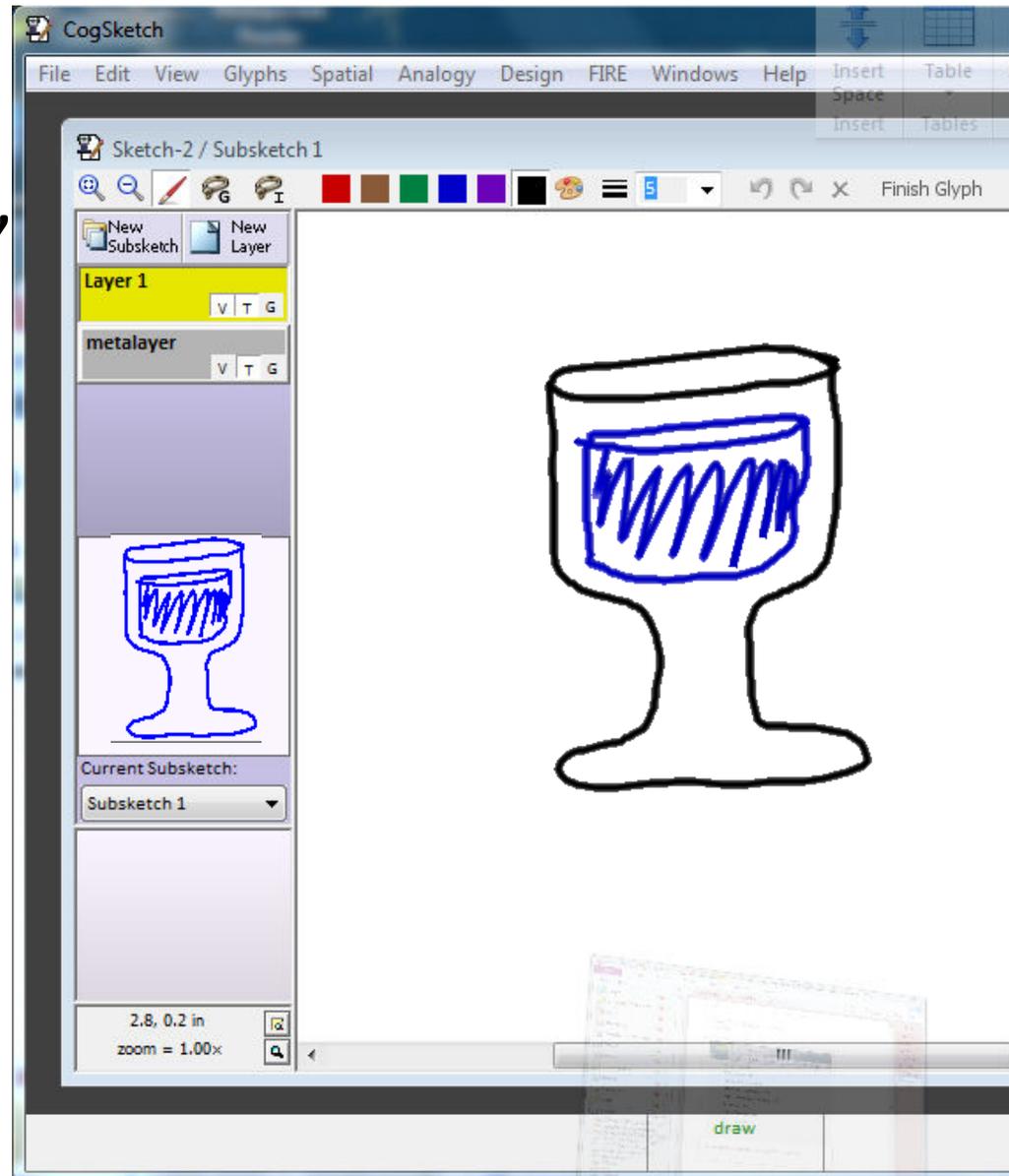
# Drawing a Glyph

- Just start drawing
- When you're done, click the finish button
- Thumbnail pane shows how ink is decomposed into glyphs via false colors



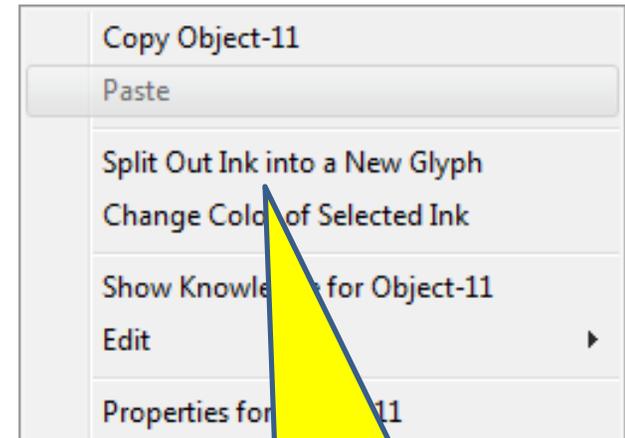
# Drawing a Glyph

- Just start drawing
- When you're done, click the finish button
- Thumbnail pane shows how ink is decomposed into glyphs via false colors

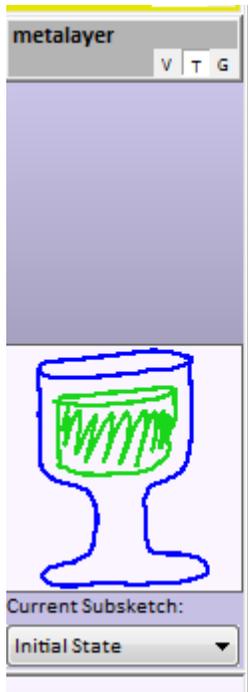


# Splitting ink into multiple glyphs

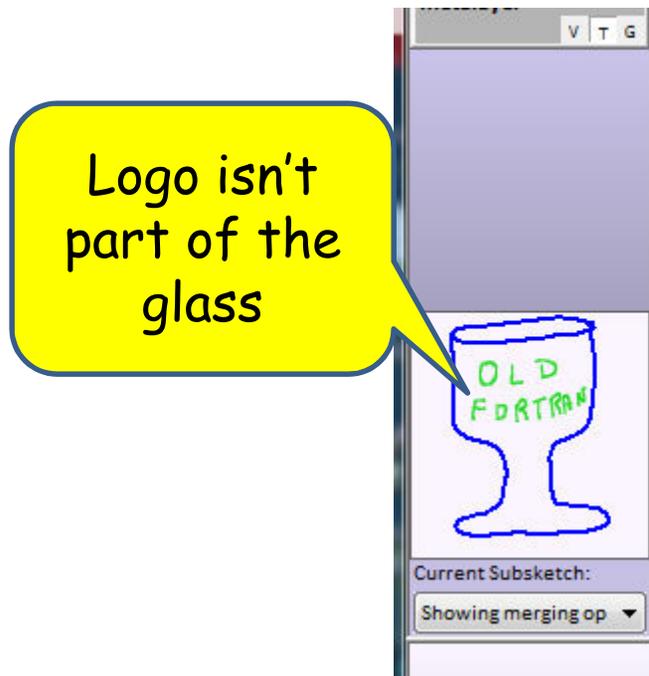
Choose the ink you want to make into a new glyph via the ink lasso



Right-click and choose split



# Merging ink into glyphs

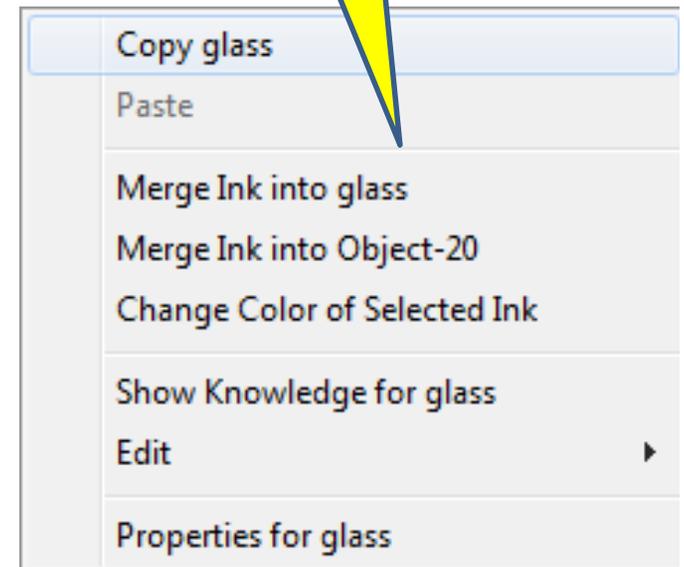


# Merging ink into glyphs

1, Select ink to be merged via the ink lasso



2. Select which is merged into the other



# Conceptual Labeling

- When people sketch, they talk
  - They say what objects are
  - They provide information that isn't easily sketched
- CogSketch provides interfaces for you to tell it what your glyphs mean
  - The most general interface is described here
  - Often simpler, customized versions are used
- The vocabulary is drawn from the OpenCyc KB contents, plus extensions
  - Concepts are defined as *collections*
  - Relationships are defined via *relations*

# Types of Glyphs

- There are three types of glyphs that you can use in CogSketch
  - **Entities:** Represent objects in a sketch. They can be concrete or abstract
  - **Relations:** Represent binary relationships between other entities in the sketch.
  - **Annotations:** Represents a property of another glyph that would be difficult to indicate in a purely visual manner

# Labeling Entities

Select the glyph to label using the Glyph lasso



Right-click and choose Properties

- Copy Selected
- Paste
- Show Knowledge for Object-11
- Edit
- Properties for Object-11

Glyph Properties: Object-11

Glyph Name: Object-11

Glyph Type: entity

What does this glyph represent?

Assigned Concepts:

- Entity

Available Concepts:

OK Cancel

Set the glyph type to Entity, if it isn't already

Glyph Type: annotation

- entity
- relation
- annotation

# Choosing a Concept

Available Concepts:

- Glacier
- Gladiator
- Gladiola
- Gladiolus
- GlagoliticLetter
- Glamour
- GlamRockBand
- GlamRockMusic
- GlamRockPerformer
- Gland
- Glanders.Infection

Start typing to see candidates

Available Concepts:

- GlassStemware
- Glass
- GlassBottle
- GlassBoxAllotment
- GlassBoxSoftwareSession-ActiveLogging
- GlassContainer
- GlassJar
- GlassStemware**
- GlassWall
- GlasswareForKitchen

Comments help indicate which choice might be best

The collection of instances of Stemware made primarily of glass.

# Choosing a Concept, continued

Glyph Properties: Object-11

 Glyph Name:  Glyph Type:

What does this glyph represent?

Assigned Concepts:

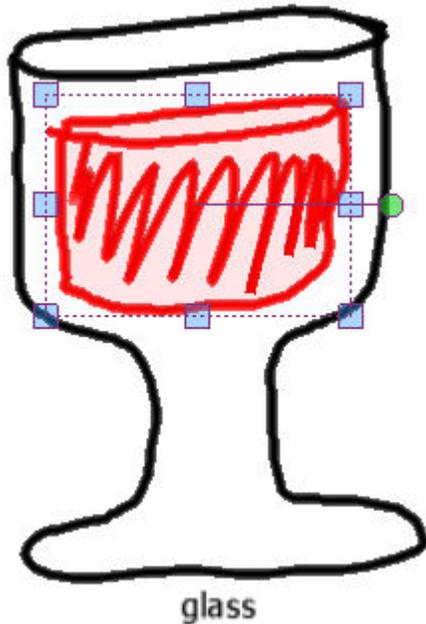
- Entity
- GlassStemware

Available Concepts:

- GlassStemware
- Glass
- GlassBottle
- GlassBoxAllotment
- GlassBoxSoftwareSession-ActiveLogging
- GlassContainer
- GlassJar
- GlassStemware**
- GlassWall
- GlasswareForKitchen

Use arrow to add selection, x to remove selection from left

# Choosing a Concept, continued



Glyph Properties: Object-12

Glyph Type: entity

Glyph Name: water

What does this glyph represent?

Assigned Concepts:

- Entity
- Water

Available Concepts:

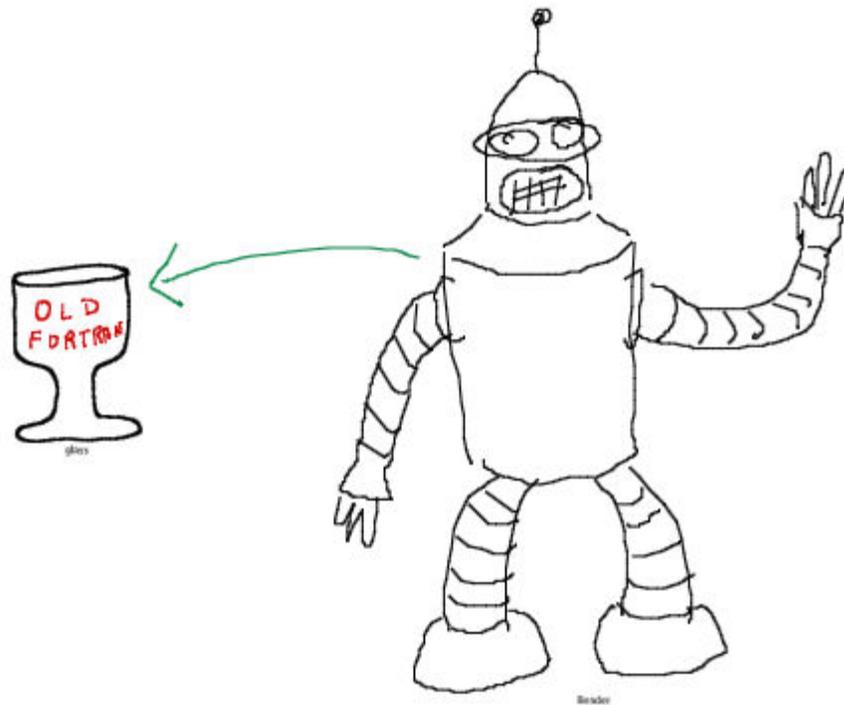
- Water
- Water-Fresh
- Water-Ingestible
- Water-Saline
- Water1
- Water2
- WaterBed
- WaterbedMarketCategory
- WaterBoard
- WaterBoardingEvent
- WaterBodyConcent

An instance of  
ChemicalCompoundTypeByChemicalSpecies and a  
specialization of InanimateThing. Each instance of  
Water is one piece of some (pure or impure) portion  
of the chemical compound H2O. Instances of Water

OK Cancel

# Relation Glyphs

- Indicate relationships between two things in the sketch
- Always drawn as arrows, as per concept maps



# Labeling a Relation Glyph

**Glyph Properties: Object-31** [Close]

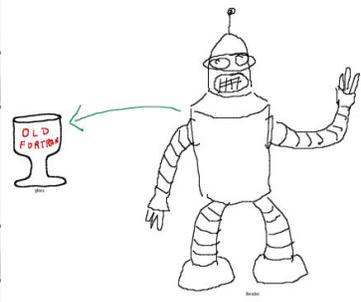
 Glyph Name:  Glyph Type:

**What relation does this glyph represent?**

**Relation Arguments:**

allow any glyph?  
<SocialBeing>

  allow any glyph?  
glass



**(owns <SocialBeing> glass)**

A predicate that relates SocialBeings to things that they own. (owns AGENT OBJECT) means that AGENT has full ownership of OBJECT. Thus, AGENT enjoys FullUseRights (q.v.) over OBJECT. OBJECT might be a physical artifact, a parcel of land, an animal, a piece of

# A fully specified relationship

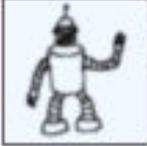
Glyph Properties: Object-31

 Glyph Name:

Glyph Type:

**What relation does this glyph represent?**

**Relation Arguments:**

  allow any glyph?

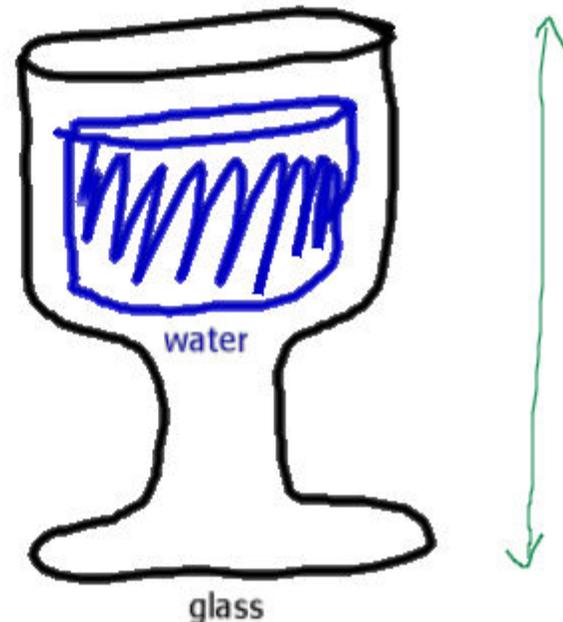
  allow any glyph?

**(owns Bender glass)**

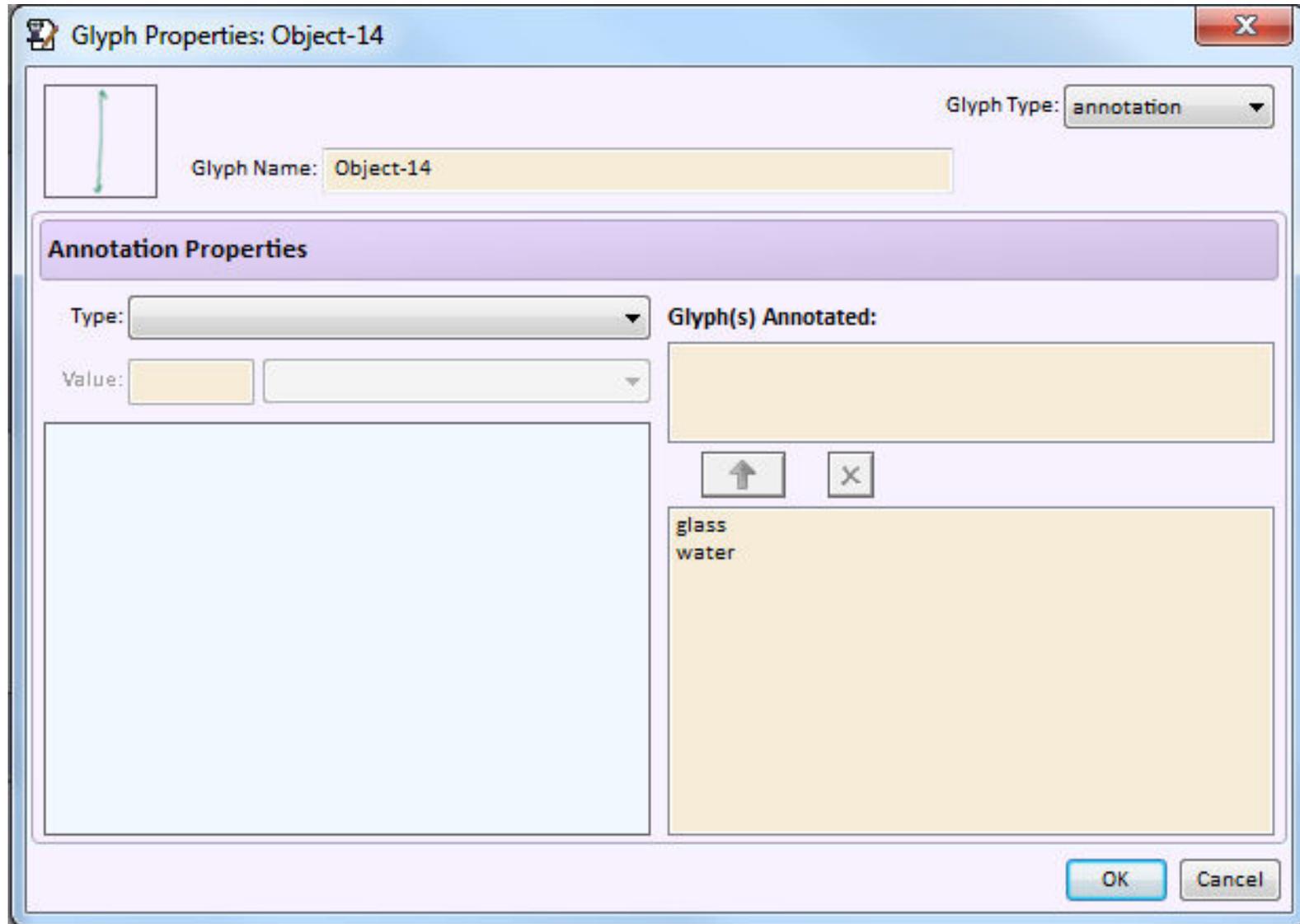
A predicate that relates SocialBeings to things that they own. (owns AGENT OBJECT) means that AGENT has full ownership of OBJECT. Thus, AGENT enjoys FullUseRights (q.v.) over OBJECT. OBJECT might be a physical artifact, a parcel of land, an animal, a piece of

# Annotation Glyphs

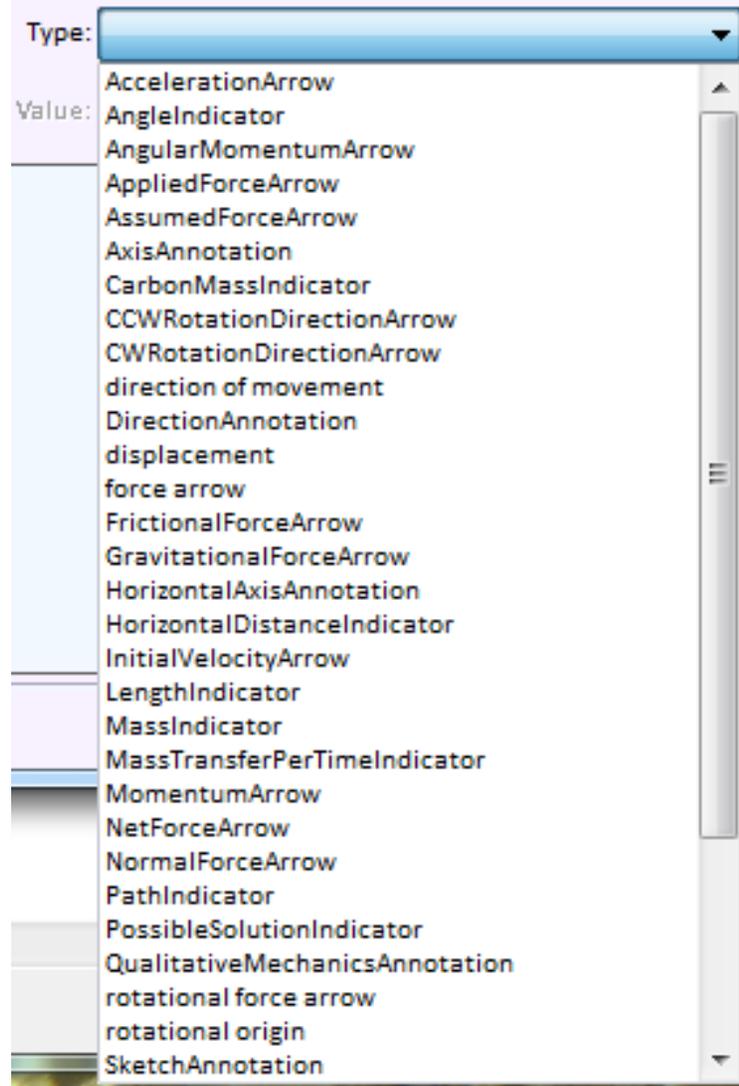
- Represent information about a glyph that would be hard to express visually
- Annotation glyph provides
  - Visual indicator in the sketch
  - Non-visual information



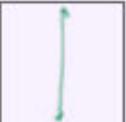
# Labeling an Annotation Glyph



# Types of Annotations



**Glyph Properties: Object-14**

 Glyph Name: 6"

**Annotation Properties**

Type: LengthIndicator

Value: 6

LengthIndicator is the used to indicate the calculated.

- Angstrom
- Attometer
- Centimeter
- Decameter
- Decimeter
- Ell
- Em
- En
- Exameter
- Fathom
- Femtometer
- Foot-UnitOfMeasure
- Furlong
- Gigameter
- Hectometer
- HundredthsOfInch
- Inch
- Kilometer
- League-Distance
- LightYear
- Megameter
- Meter
- Micrometer
- Micron
- Mile
- Millimeter
- Nanometer
- NauticalMile
- NucleotideLong
- NucleotidePairLong

The glyph name is only for the user. CogSketch does not try to interpret the glyph name.

Values are optional, and their meaning depends on the annotation type

**Glyph(s) Annotated:**

glass

↑ ×

water

You choose what is being annotated

**Glyph Properties: Object-14**

Glyph Type: **annotation**

Glyph Name: **6"**

**Annotation Properties**

Type: **LengthIndicator**

Value: **6** **Inch**

LengthIndicator is the collection of sketch annotations used to indicate the length between two points should be calculated.

Glyph(s) Annotated:

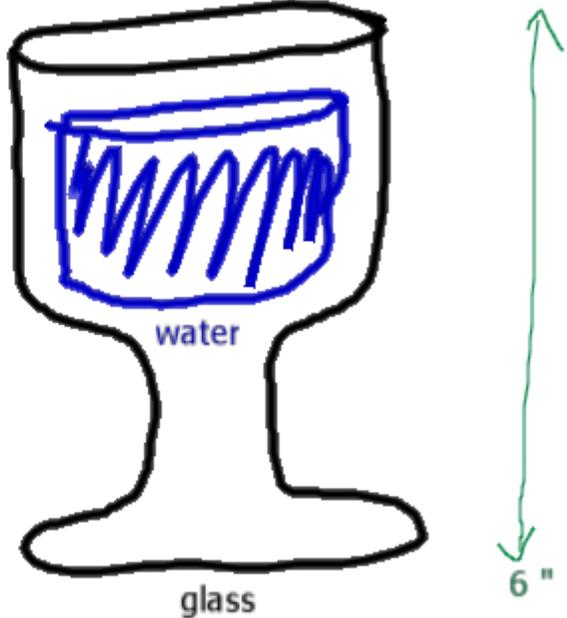
- glass
- water

↑ ×

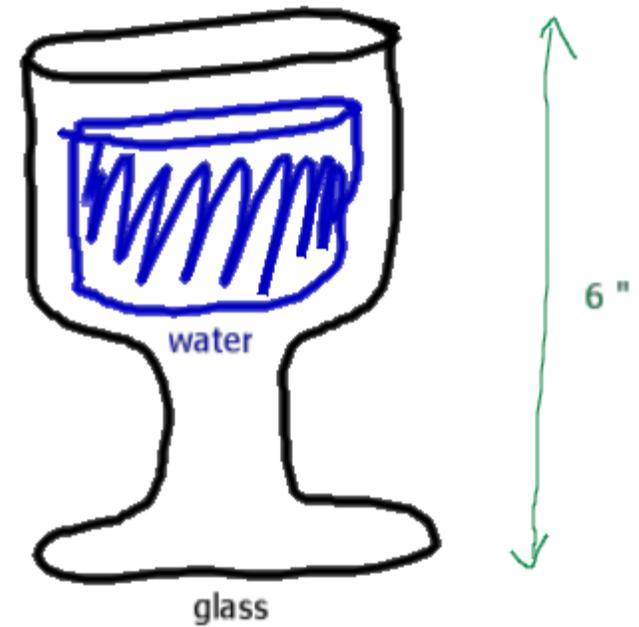
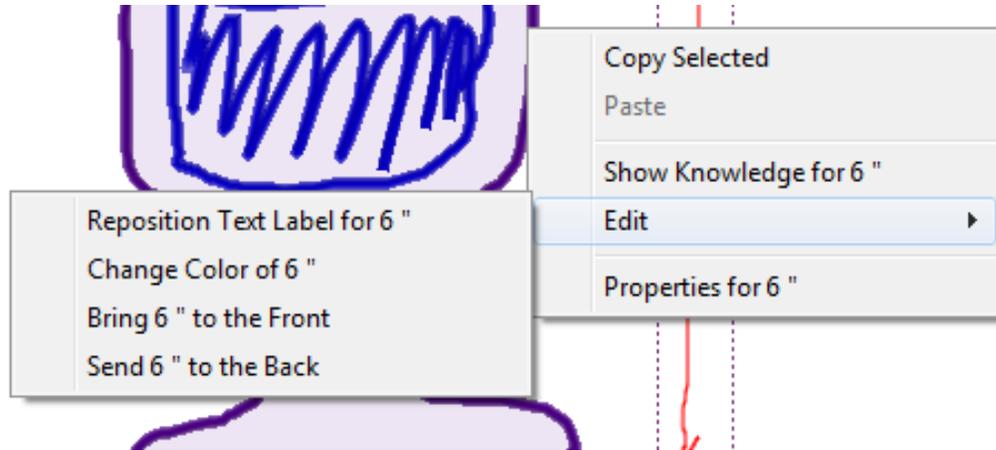
**metayer** V T G



Current Subsketch: **Initial State**



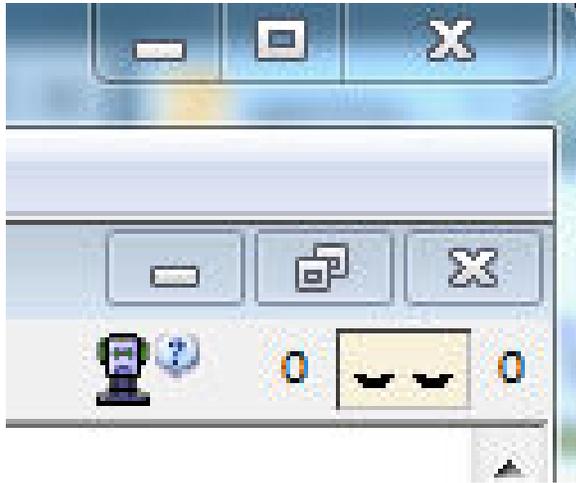
# Neatening your Sketch



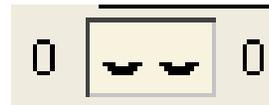
# What CogSketch does with this information

- It enables CogSketch to reason about the objects you sketched
  - e.g. simple qualitative mechanics is built-in
  - Used in Design Coach, described later
- It enables CogSketch to match sketches
  - e.g., sketch worksheets for education compare a student's sketch with a teacher's sketch
  - Understanding intended meaning of glyph via labeling is vital because students are often incorrect

# Status Indicators



Waiting. Happens while you are drawing/moving something, postponing visual processing until you are finished, for responsiveness.



Idle. Digits indicate the number of tasks queued



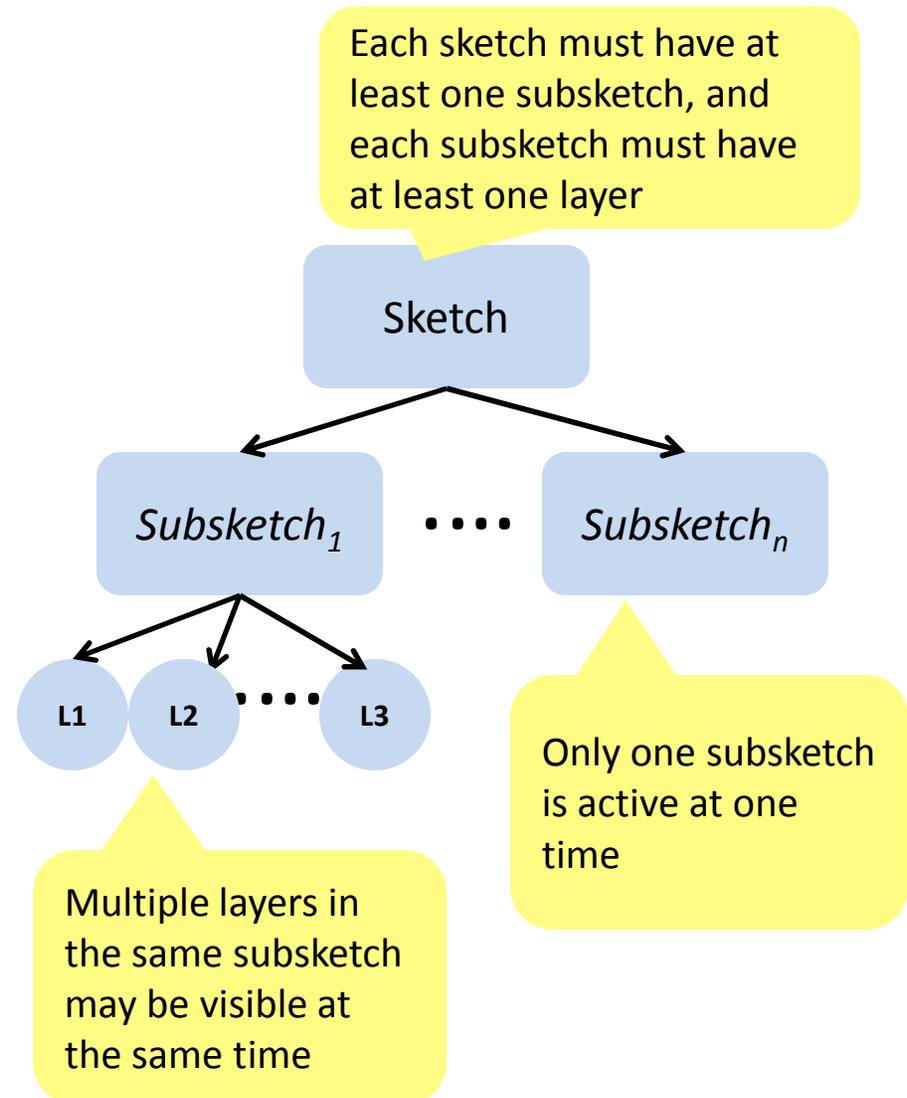
Both processors running



Crashed. (Very rare) Touch to restart

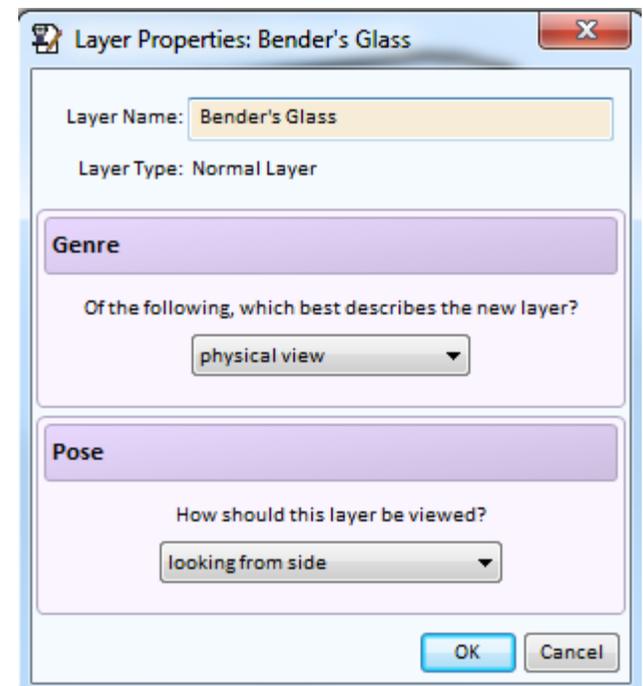
# The structure of sketches

- Think of layers as transparent sheets stacked on top of each other
  - Multiple layers in the same subsketch can be visible at the same time
  - Spatial relationships will only be computed between objects on the same layer



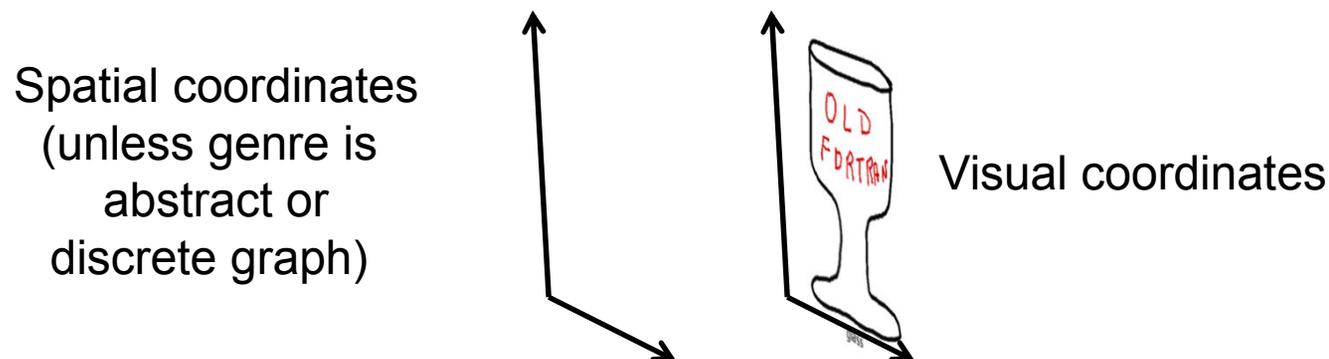
# How Layers are Interpreted

- Every layer has a *genre* and *pose*
- *Genre* indicates the kind of sketch it is
- *Pose* concerns frame of reference, defining how visual properties map to spatial properties



# Genres

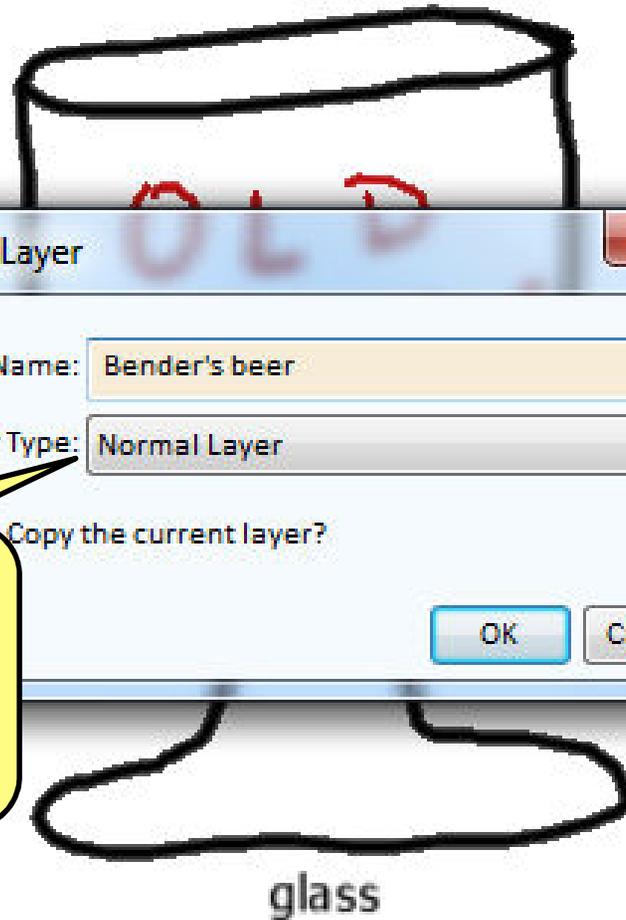
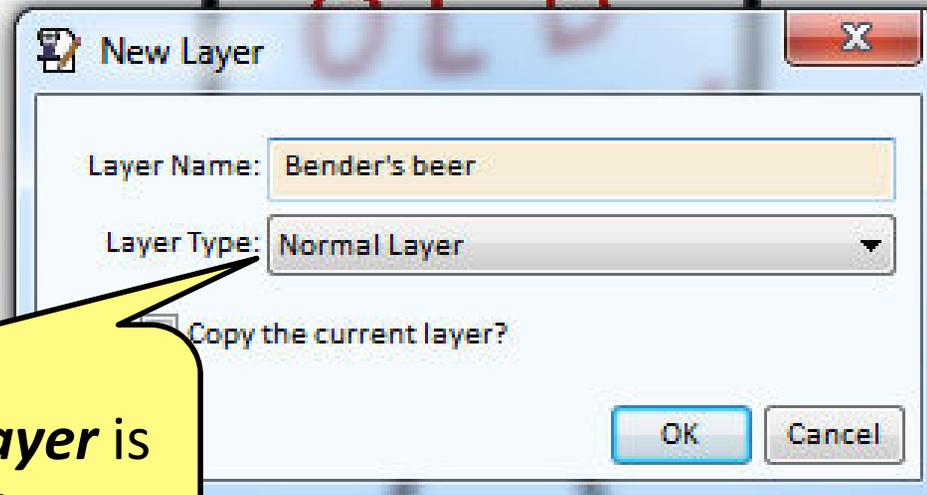
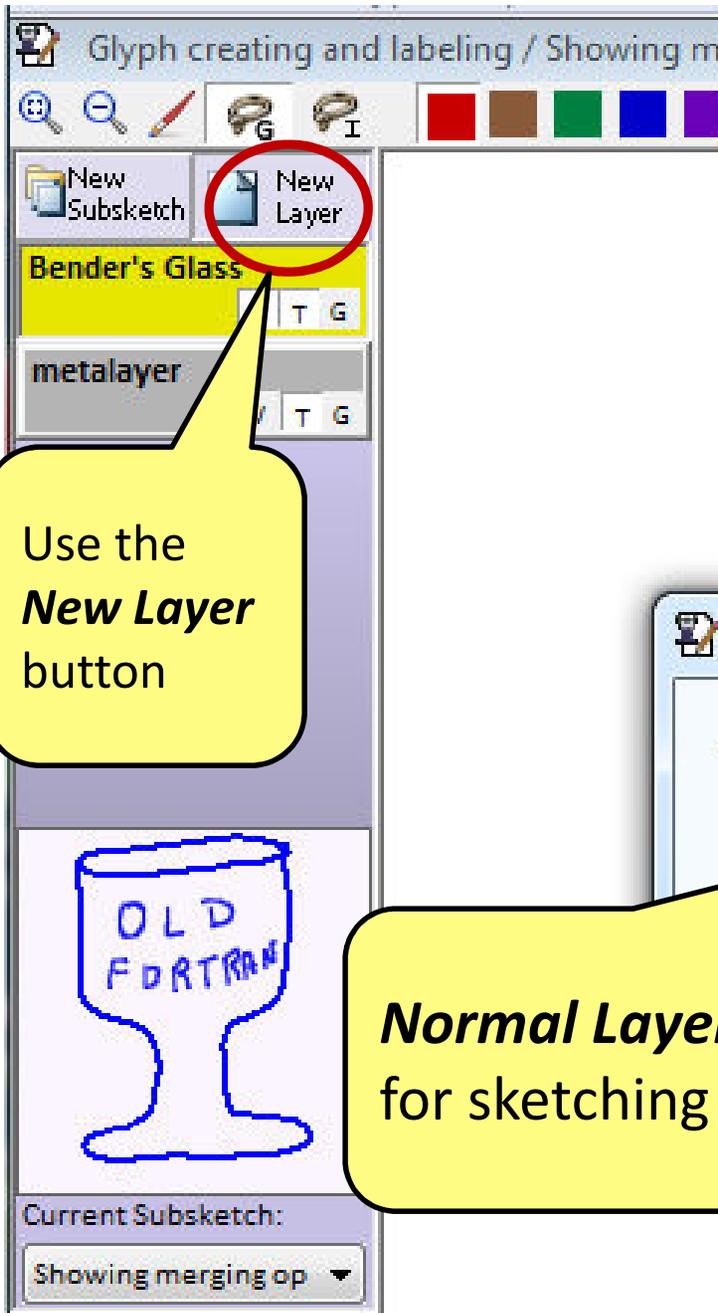
- *Abstract*: Visual relationships between glyphs provide no information about spatial relationships between them
- *Discrete graph*: Visual contact relationships important, but other visual properties (e.g. distances and locations) are not
  - Example: Concept maps
- *Geospatial*: Visual coordinates map onto geospatial coordinates, direction into N/S/E/W
- *Physical*: Visual coordinates map onto spatial coordinates, spatial relations are up/down/left/right



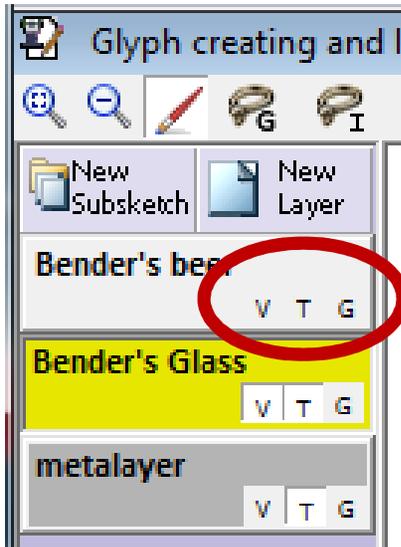
# Pose

- *Unspecified*: Holds for abstract and discrete graph genres only.
- *Looking from bottom*: Up vertical = from user into the sketch.
- *Looking from top*: Up vertical = from the sketch to the user.
- *Looking from side*: Up vertical = up in glyph space

# Adding a Layer



# Controlling Visibility



- V = Visible or not
- T = show text labels or not
- G = Grey out layer contents



glass



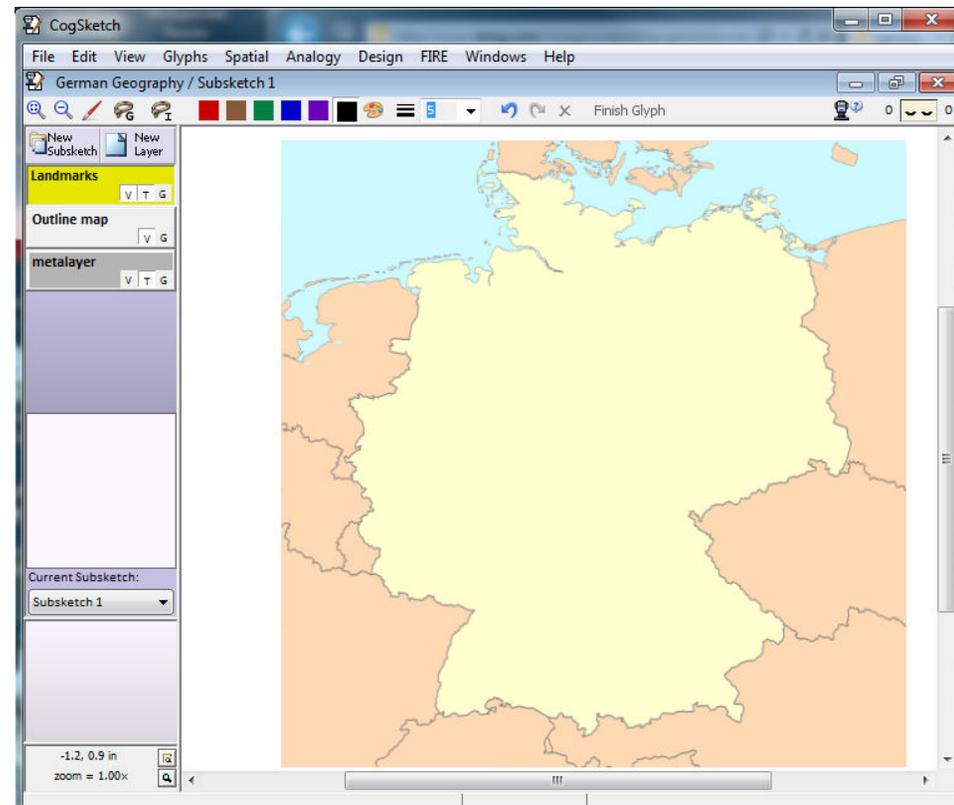
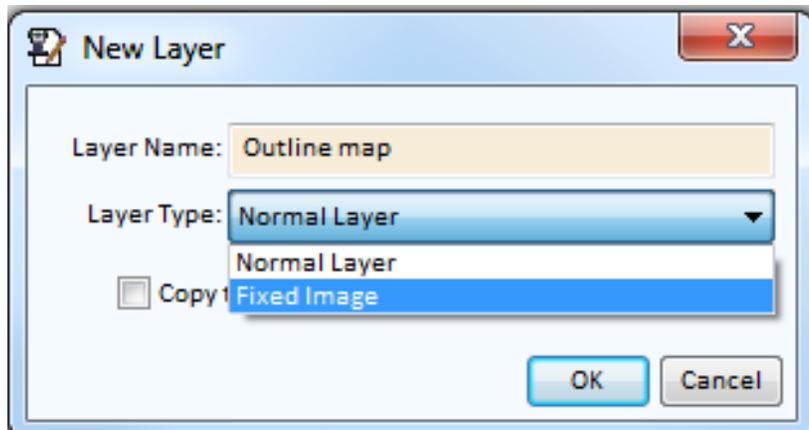
glass



glass

# Adding a Bitmap Layer

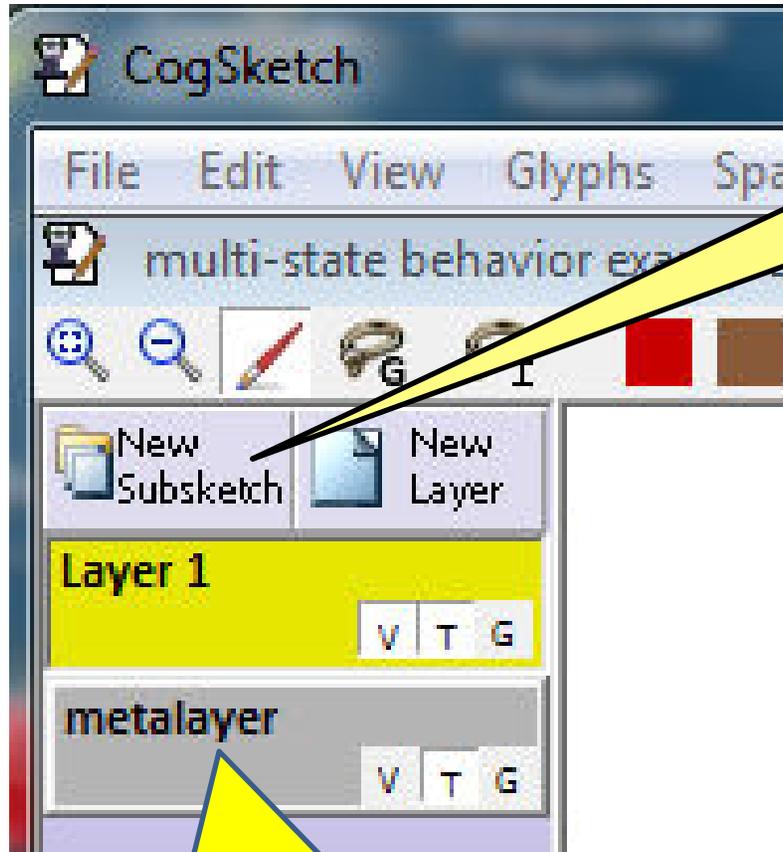
- Useful for providing something to draw on top of
  - Annotating photographs or diagrams is a common task for sketch worksheets



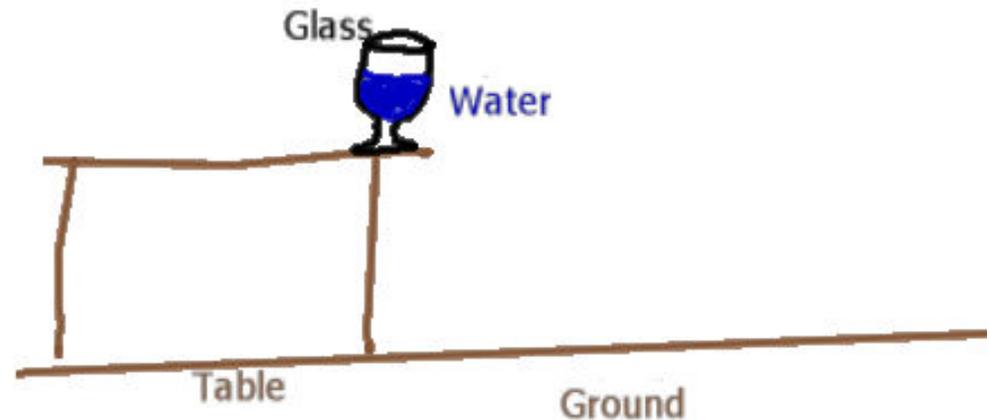
# Uses for Multiple Subsketches

- Describing a complex behavior
  - Each subsketch might represent a distinct qualitative state
  - Can create *comic graphs*, a generalization of comic strips, that allow branches and joins in addition to sequentiality
- Describing alternatives
- Describing something from multiple perspectives

# Adding a Subsketch



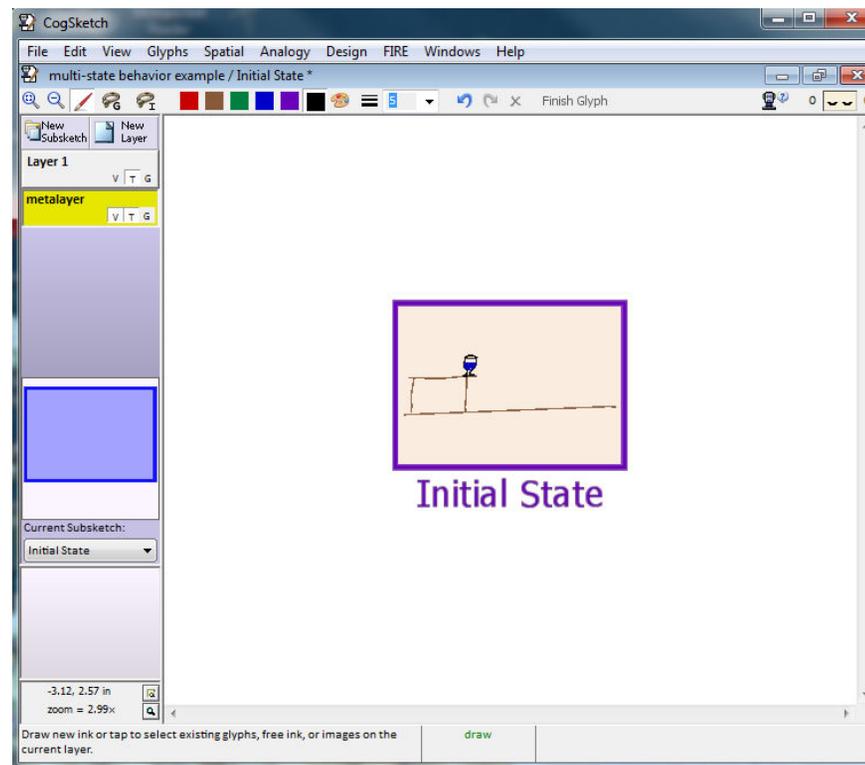
You can add a subsketch using the ***New Subsketch*** button



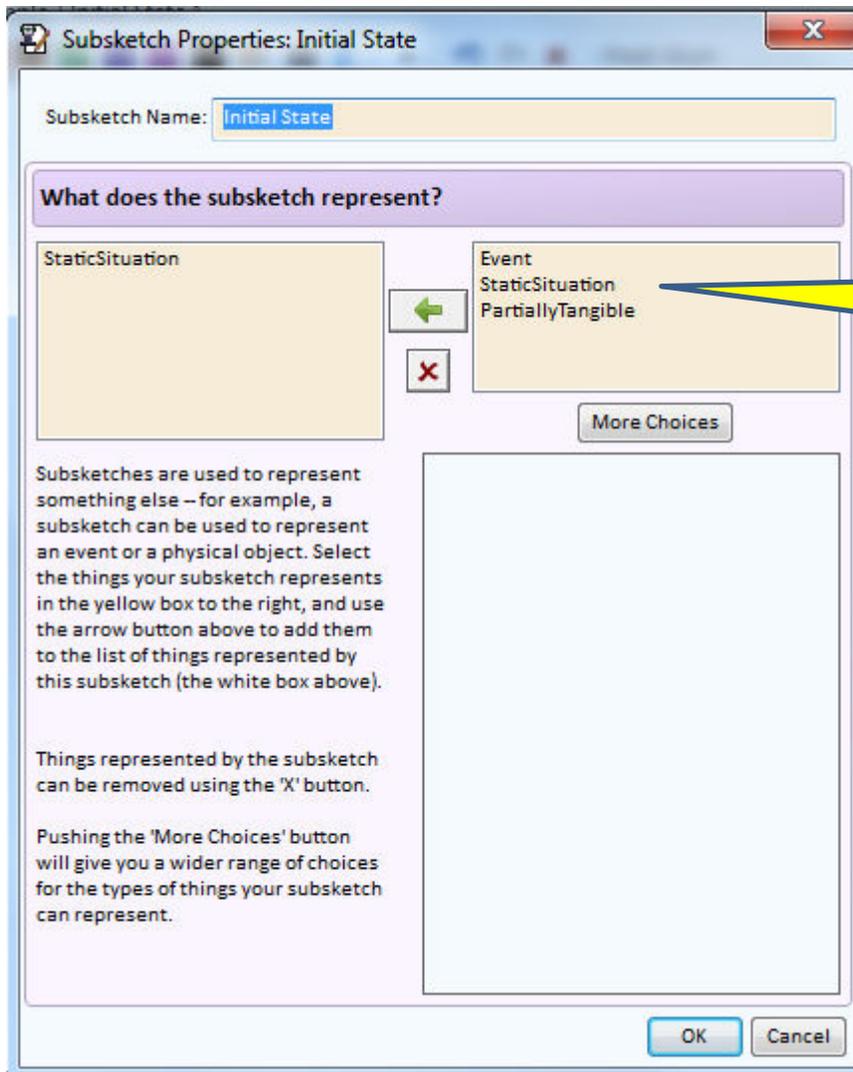
The *metalayer* lets you manipulate subsketches

# The Metalayer

- Every subsketch is a glyph on the metalayer
- Subsketch glyphs can be connected via relation glyphs, and annotated

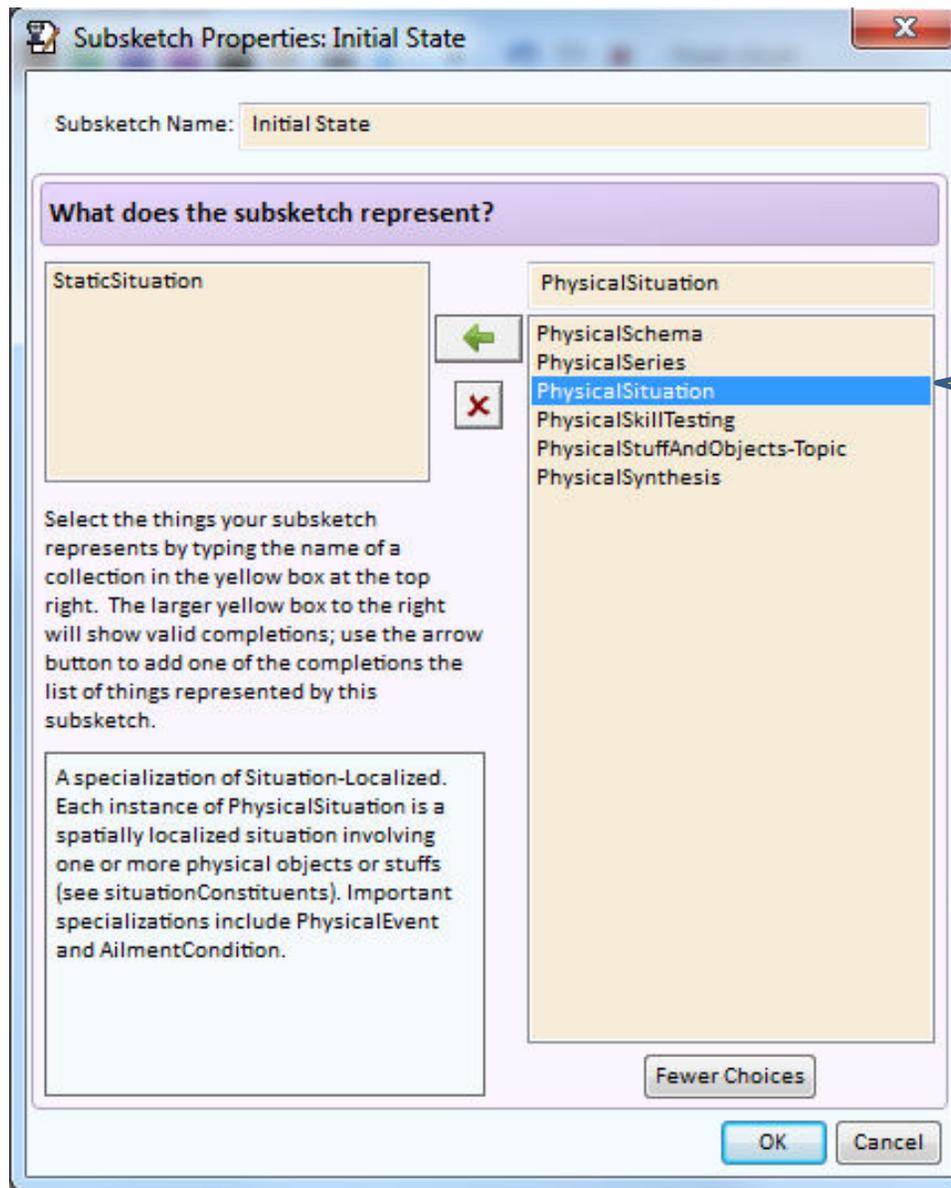


# Conceptually Labeling a Subsketch



Default = a static configuration

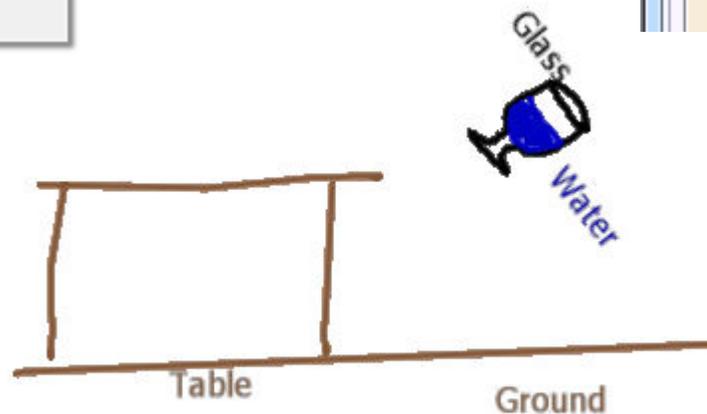
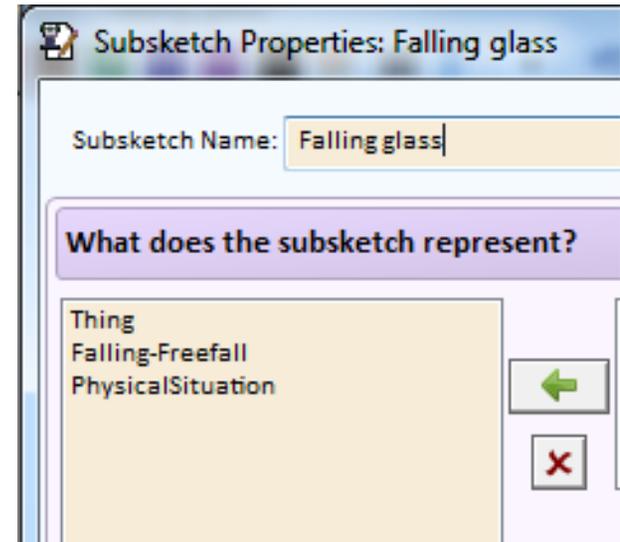
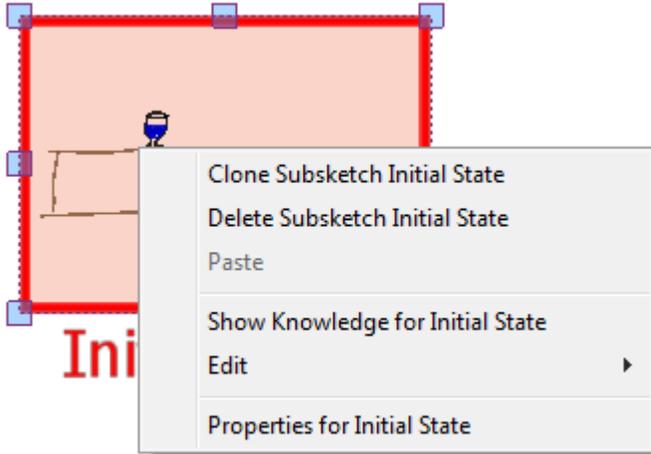
# Conceptually Labeling a Subsketch



You can indicate that the subsketch is an instance of something else via selecting a different concept

# Cloning

- Easiest way to rapidly describe complex behaviors
  - Clone subsketch, then modify the clone appropriately
  - Add arrows to indicate how they are related



# Linking the Behaviors

**Glyph Properties: Object-39**

Glyph Type: relation

Glyph Name: after

What relation does this glyph represent?

after-Underspecified

after-Underspecified

Relation Arguments:

allow any glyph?  
Falling glass

allow any glyph?  
Initial State

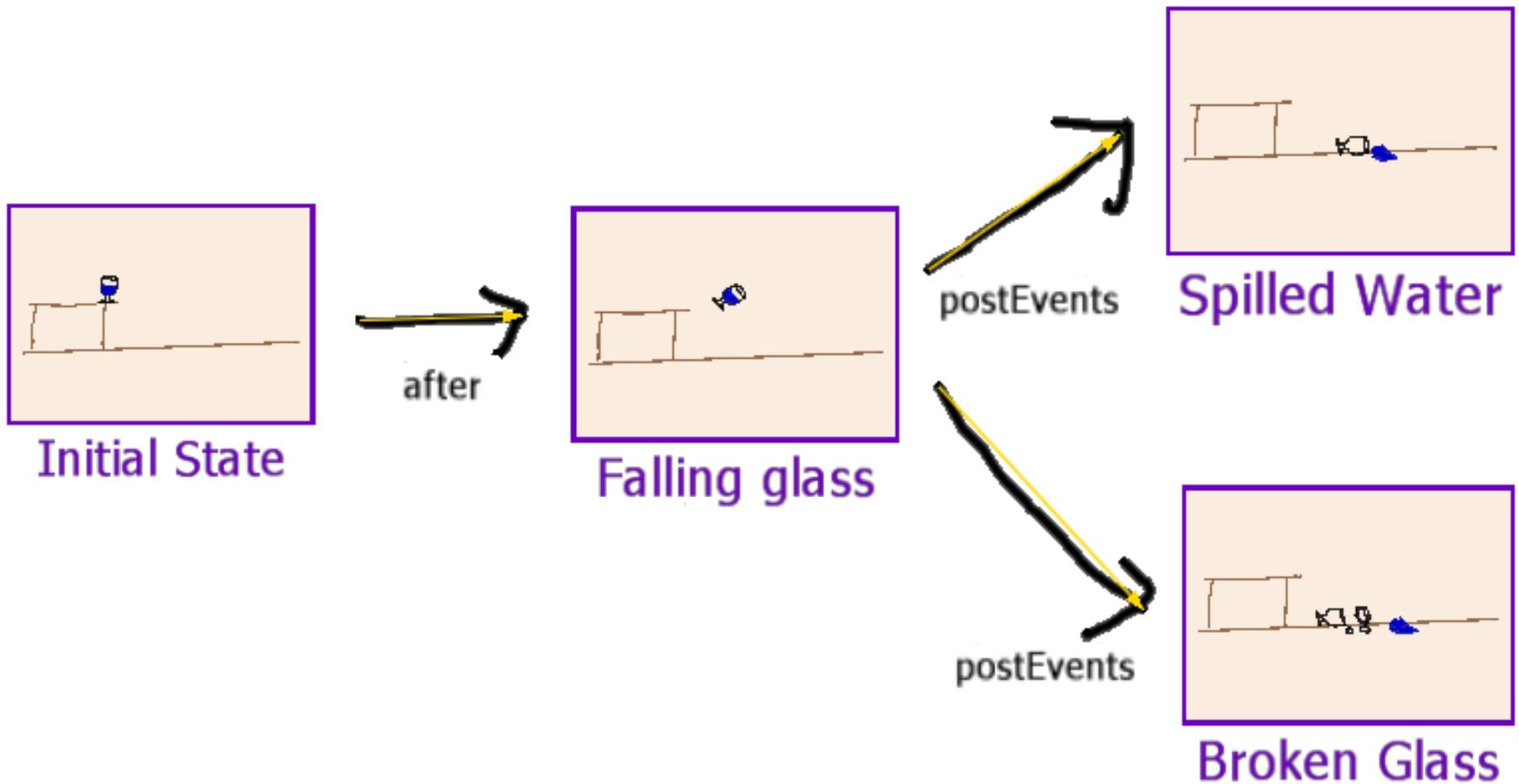
**(after-Underspecified Falling glass Initial State)**

An UnderspecifiedPredicate. (after-Underspecified THING1 THING2) means that THING1 is in some sense after THING2.

OK Cancel

The diagram to the right of the dialog shows two boxes representing states. The left box is labeled 'Initial State' and the right box is labeled 'Falling glass'. A yellow arrow points from the 'Initial State' box to the 'Falling glass' box, with the word 'after' written below the arrow.

# A Comic Graph



# What you have seen

- Sketches are made of glyphs
  - How to draw glyphs
  - Types of glyphs: Entities, relations, annotations
- Structure of sketches
  - Layers, subsketches, and the metalayer