

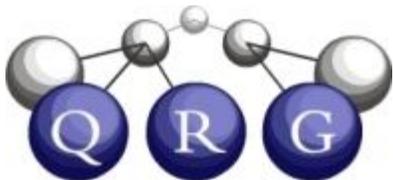
# CogSketch Tutorial

Ken Forbus

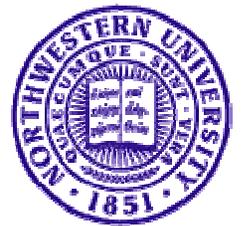
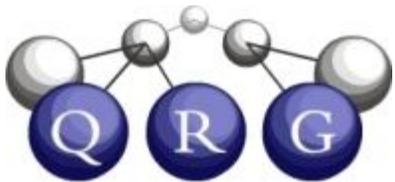
Matt McLure

Maria Chang

Northwestern University



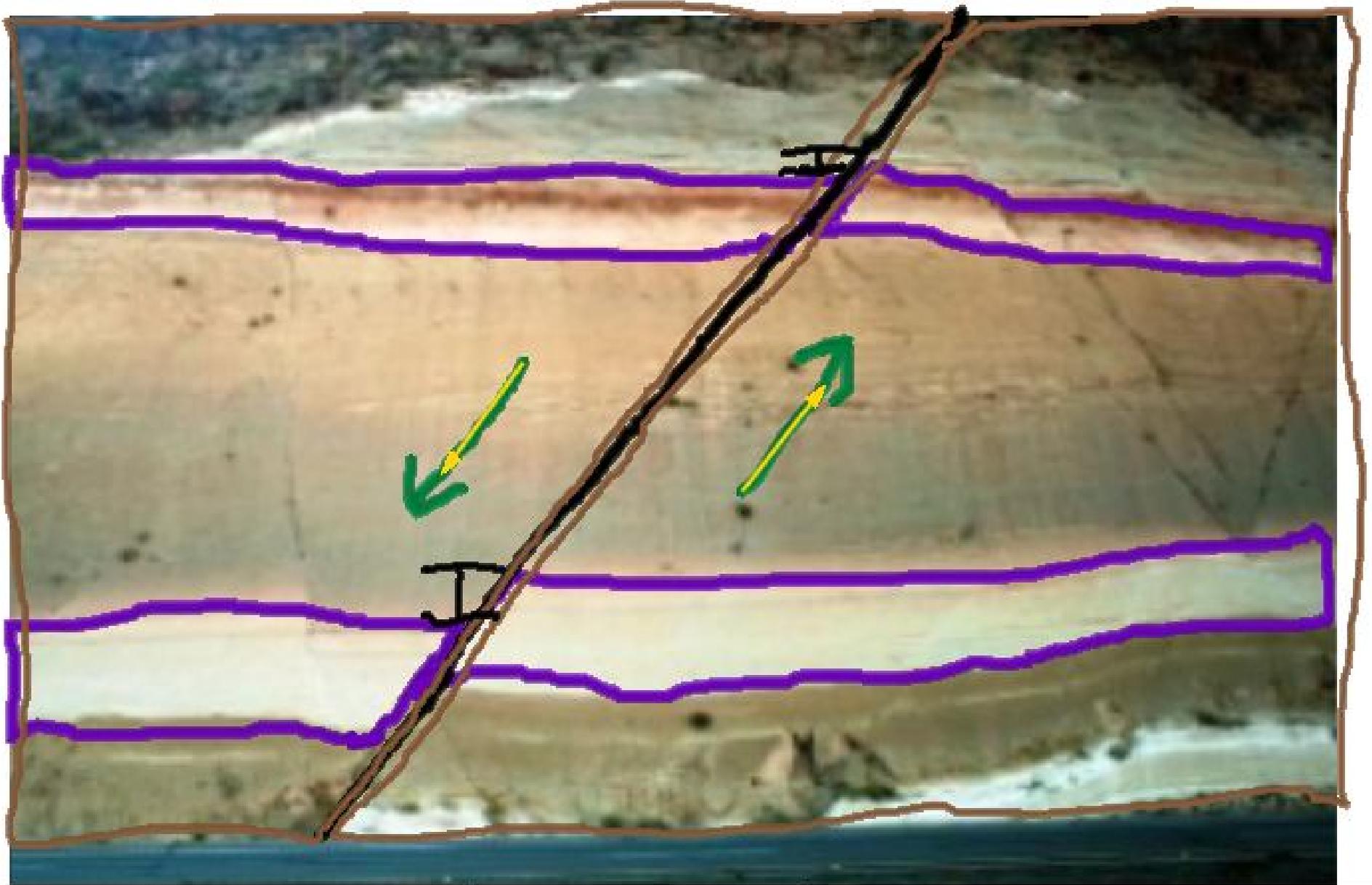
# Welcome



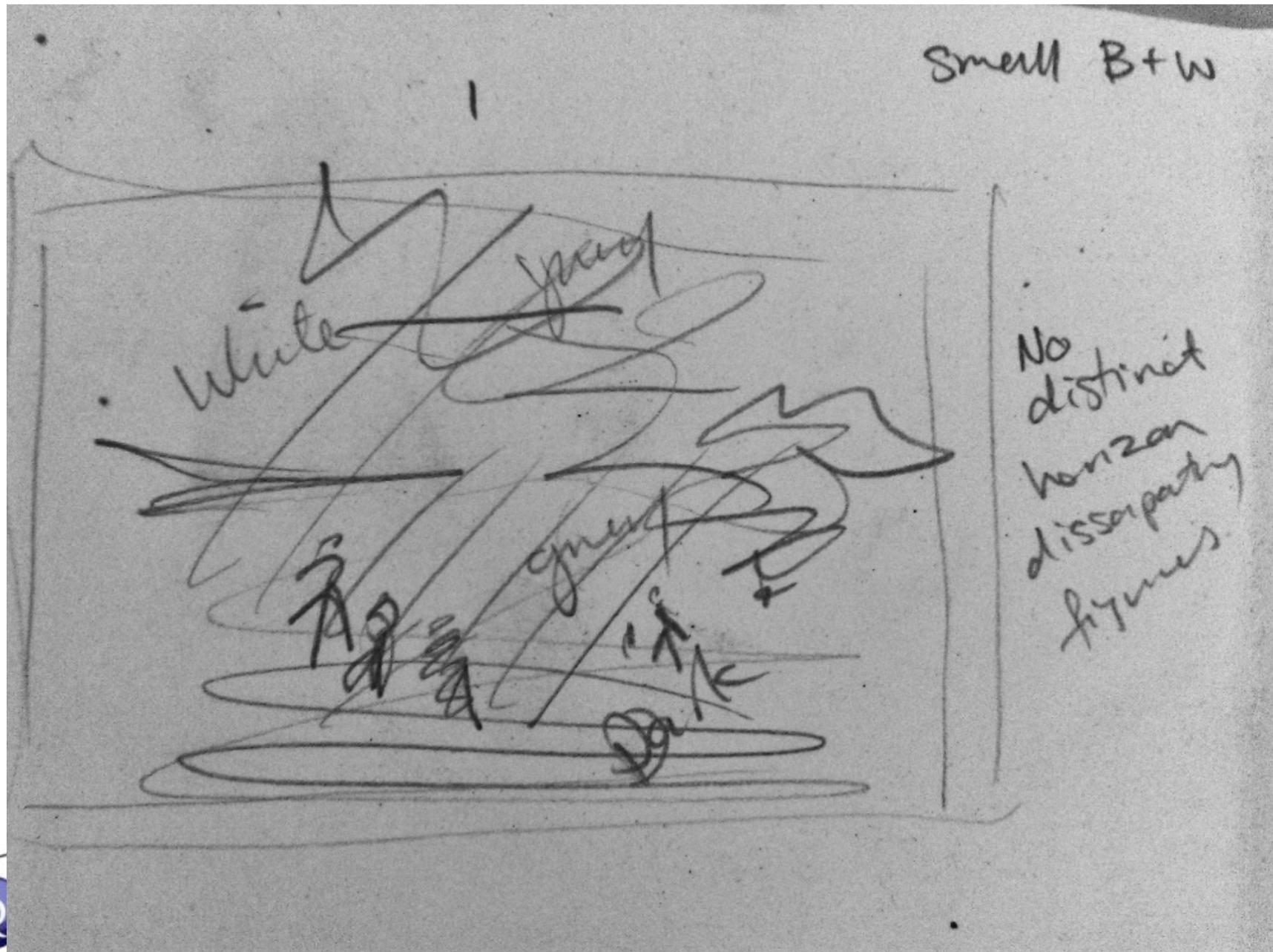
# Sketching is a Form of Communication



# Sketching is a Form of Communication



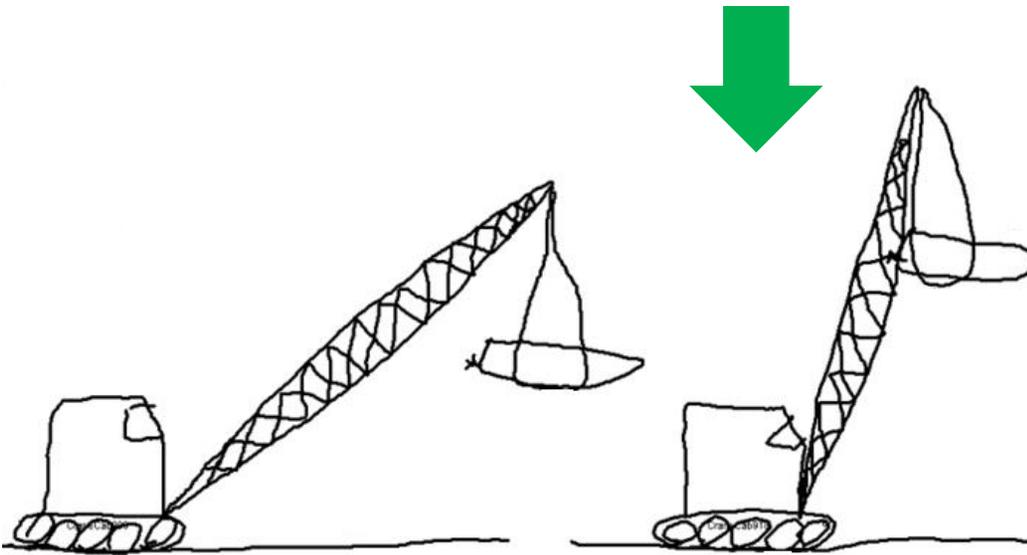
# Sketching is an Aid to Thinking



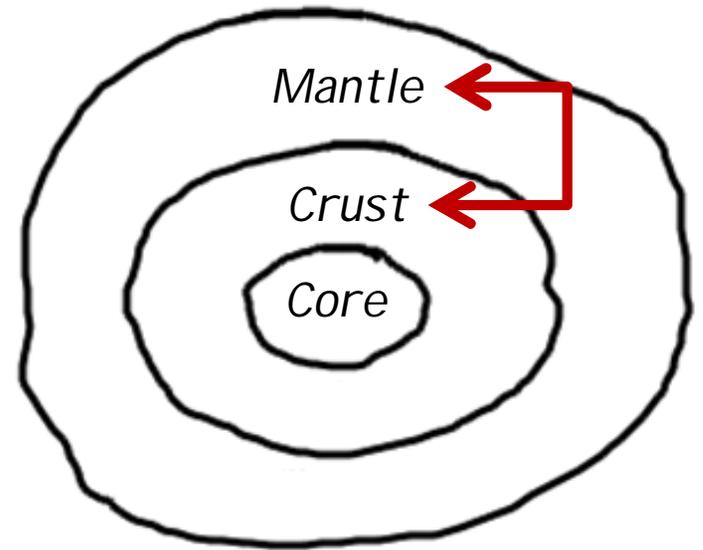
# Sketching is an Aid to Thinking



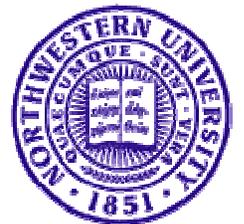
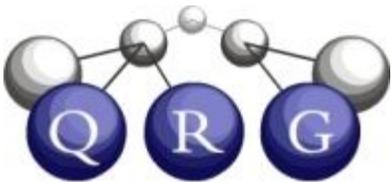
# Understanding Sketches is a Deep Scientific Problem



*Which crane is more stable?*

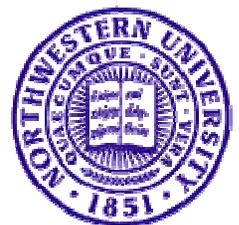


*Does this accurately depict the insides of the Earth?*



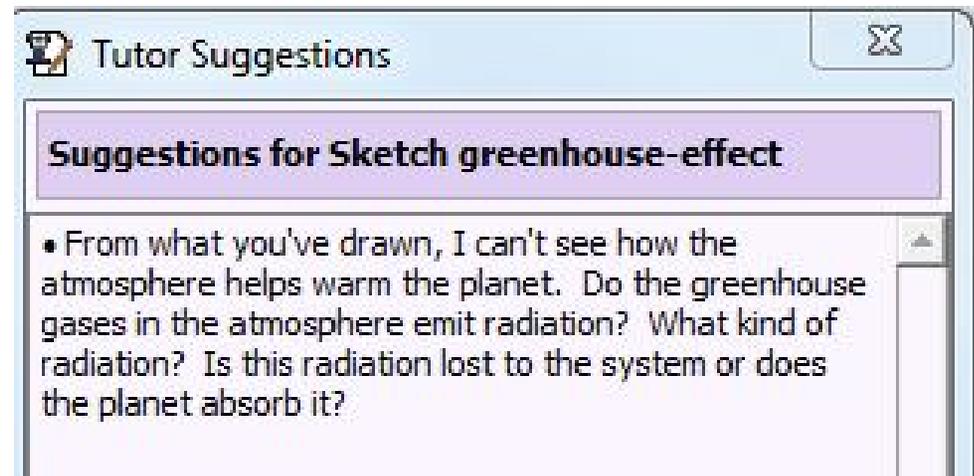
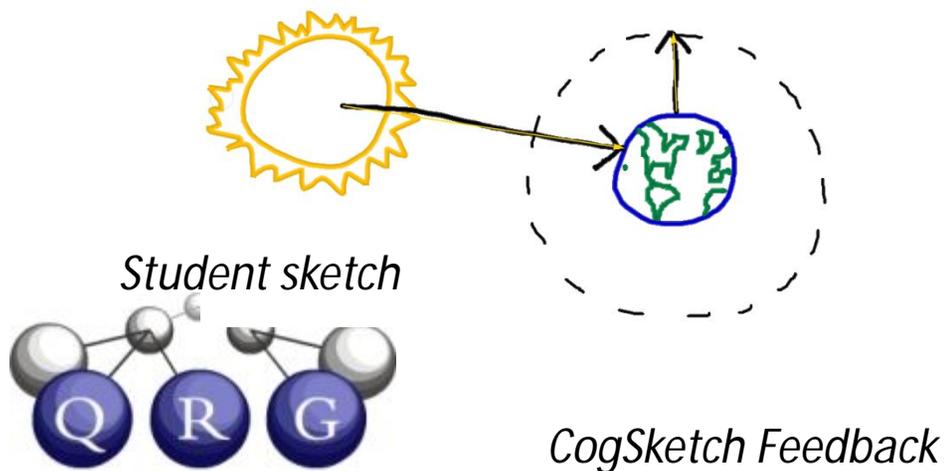
# CogSketch Goal 1: Cognitive Science Research Instrument

- Modeling human spatial reasoning and learning
  - Cognitive simulations provide new insights into human processing
  - Support AI research
- Gathering and analyzing data in laboratory and classroom studies
  - Digital ink provides time-stamped data
  - Human-like visual processing could support automated data analysis



# Computer Tutors Need Spatial Capabilities

- Intelligent tutoring systems have provided valuable benefits for education (e.g. Cognitive Tutors)
  - Immediate feedback, potentially any time, anywhere
  - Potential for large-scale assessment
- But not in spatially rich STEM subjects
  - e.g., geoscience, engineering
- Sketch understanding software could change this



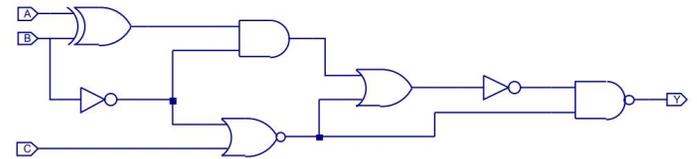
# CogSketch Goal 2: Platform for Sketch-based Intelligent Educational Software

- Focus on helping students learn STEM concepts
- Explore two models of intelligent educational sketching software
  - Sketch Worksheets, Design Coach
- **Vision: Sketch understanding software to help students learn could be widely available**



# Sketch Recognition Systems

- Goal: Provide natural, fluent interaction
- Assumes small set of visual symbols/shapes suffice to express everything

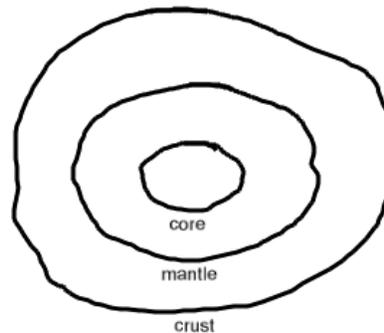
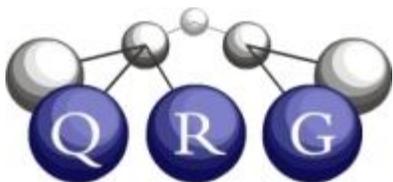


System	Problem
Newton's Pen	Draw free-body diagrams
Kirchoff's Pen	Draw resistor networks
OrganicPad, ChemPad	Draw 2D molecules, converts to 3D
Mechanix	Draw trusses, get feedback
PhysicsBook	Draw simple mechanics problems
LogiSketch	Draw logic circuits, do simulation
MathPad2	Draw equations, do animations



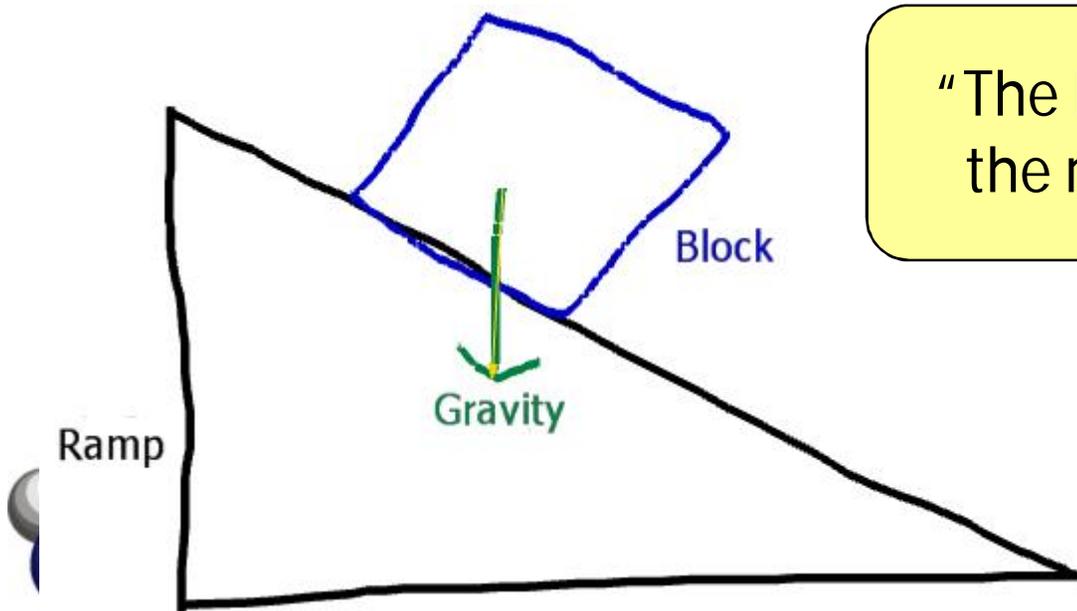
# Our approach: Open-domain sketch understanding

- Object recognition is not necessary
  - People talk when they sketch – they label objects
  - CogSketch enables people to label as they draw, avoiding the recognition bottleneck
- CogSketch models aspects of human visual and spatial representations and reasoning
  - Derives rich relational representations
  - Same software operates across many domains

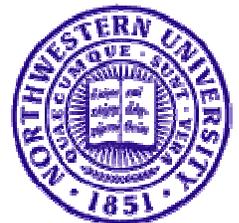


# Sketching: A Long-Term Vision

- Software that understands sketches as people do
  - Fluent, natural interaction
  - Human-like visual and spatial reasoning
  - Conceptual reasoning about the sketch
  - Domain-general



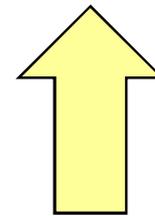
"The block will slide to the right and down"



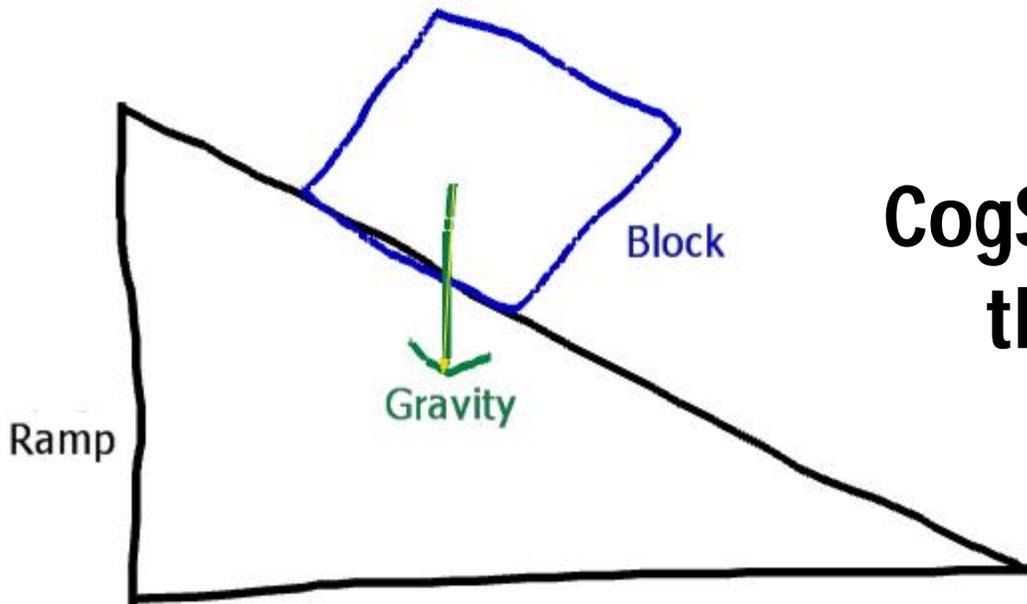
# Sketching: A Long-Term Vision

- Software that understands sketches as people do
  - Fluent, natural interaction
  - Human-like visual and spatial reasoning
  - Conceptual reasoning about the sketch
  - Domain-general

Sketch Recognition  
Research

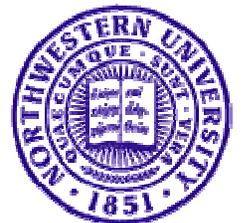


**CogSketch explores  
these aspects**

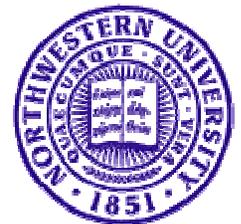
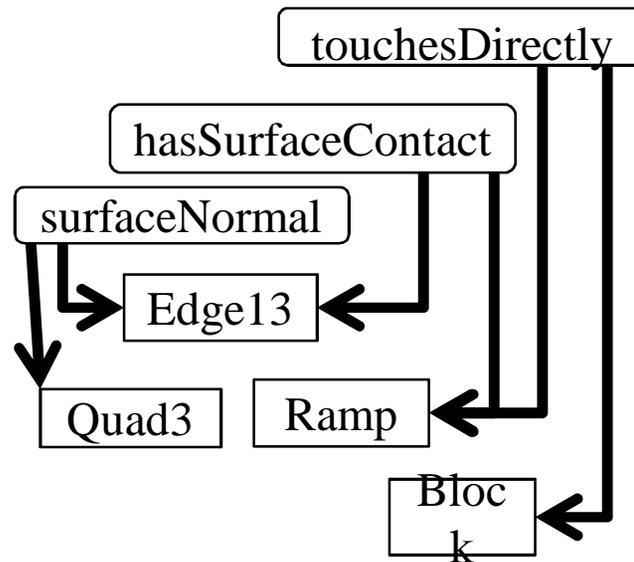
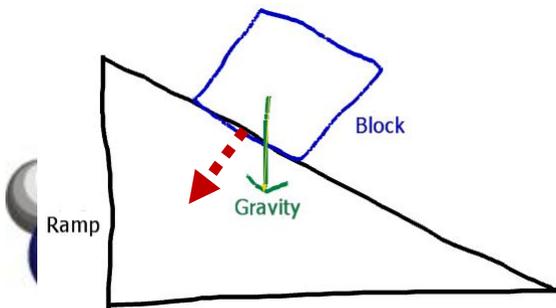
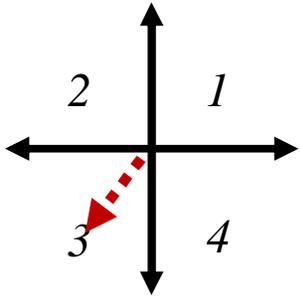


# Ideas Underlying CogSketch

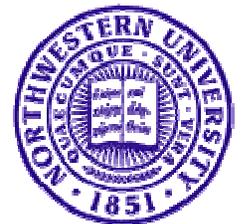
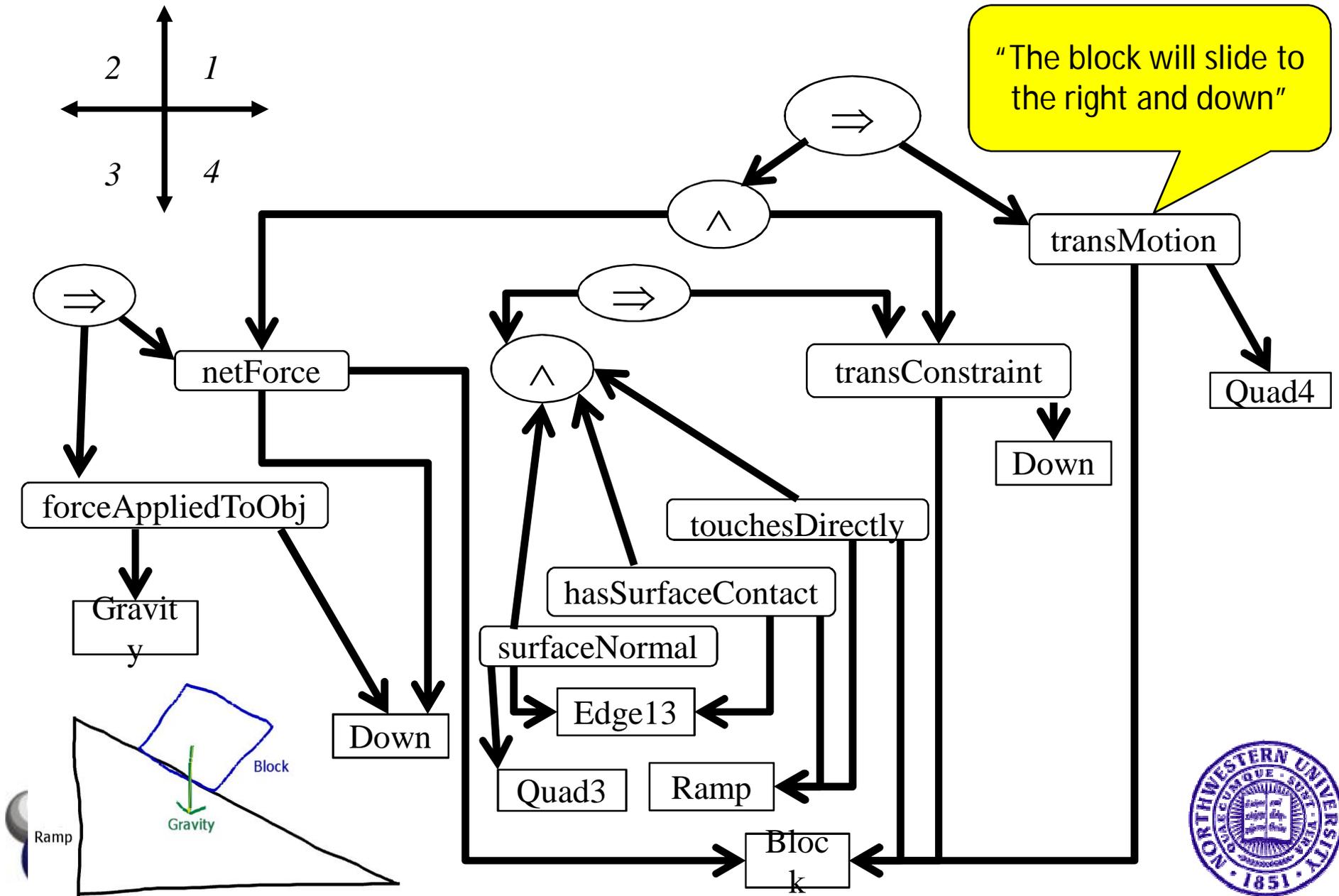
- Perceptual processing produces qualitative spatial representations
  - Forbus (1980); Huttenlocher, Hedges, & Duncan (1991)



# CogSketch Uses Visual Reasoning

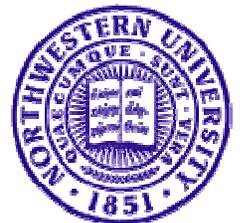


# CogSketch Uses Qualitative Reasoning



# Ideas Underlying CogSketch

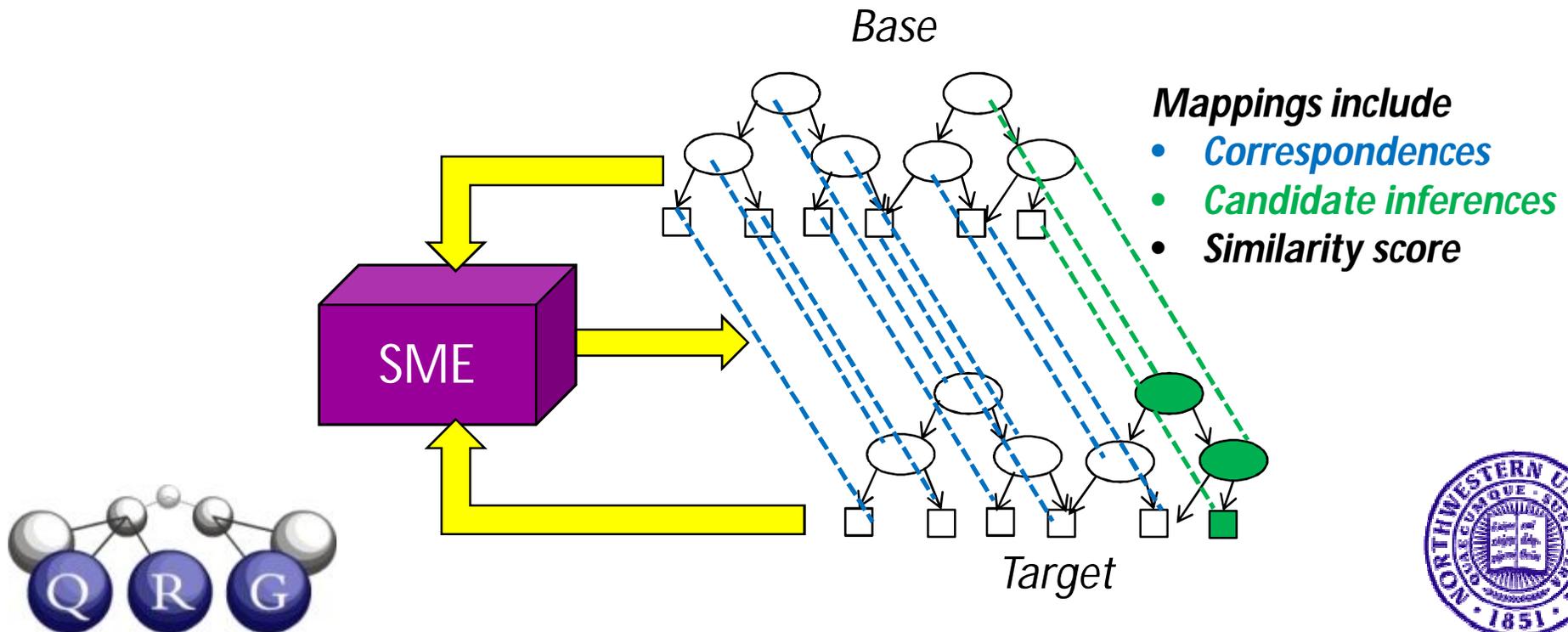
- Perceptual processing produces qualitative spatial representations
  - Forbus (1980); Huttenlocher, Hedges, & Duncan (1991)
- Structure-mapping processes are used in visual reasoning
  - Lovett, Gentner, Forbus, & Sagi, 2009; Markman & Gentner (1996)



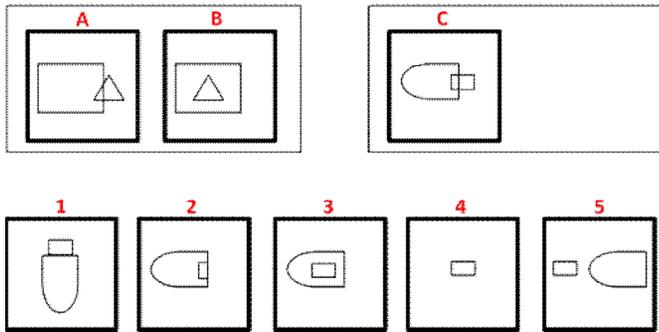
# Structure-Mapping Engine (SME)

(Falkenhainer, Forbus, & Gentner, 1986; Forbus, Ferguson, & Gentner, 1994)

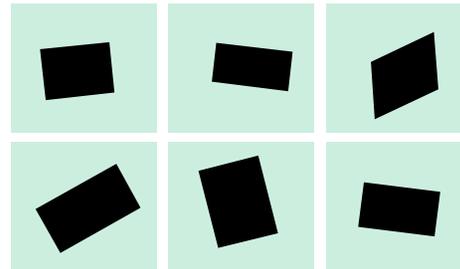
- Model of analogical comparison
  - Based on Gentner's (1983) Structure-Mapping Theory
- Compares cases by aligning their common structure



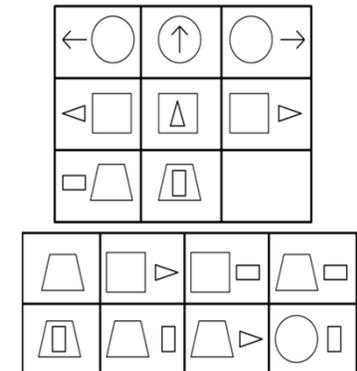
# CogSketch Visual Modeling



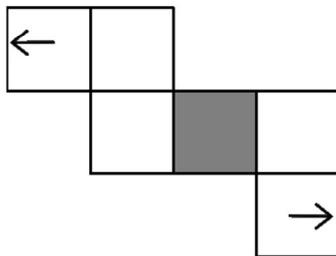
*Predicts human reaction time differences in geometric analogies*



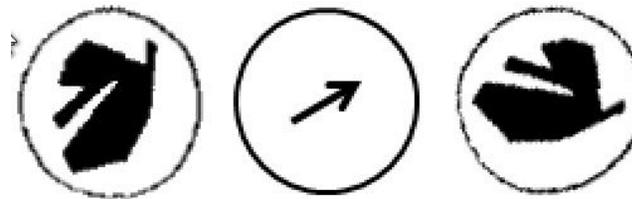
*Captures cross-cultural effects in visual oddity task*



*76<sup>th</sup> percentile, better than most adult Americans, at Raven's Progressive Matrices*



*Handles all 10 classes of paper-folding tasks*

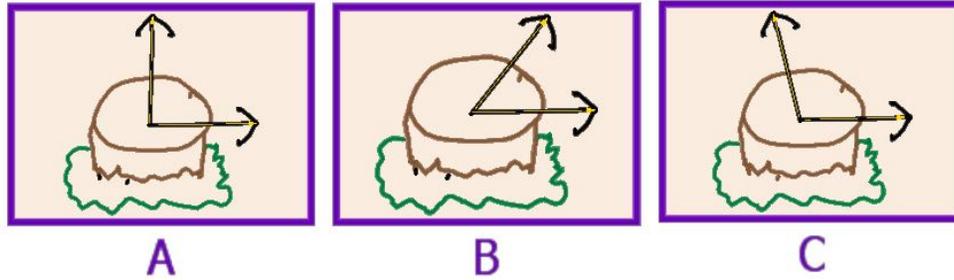


*Suggests spatial abstraction strategies as explanation for variations in human performance*

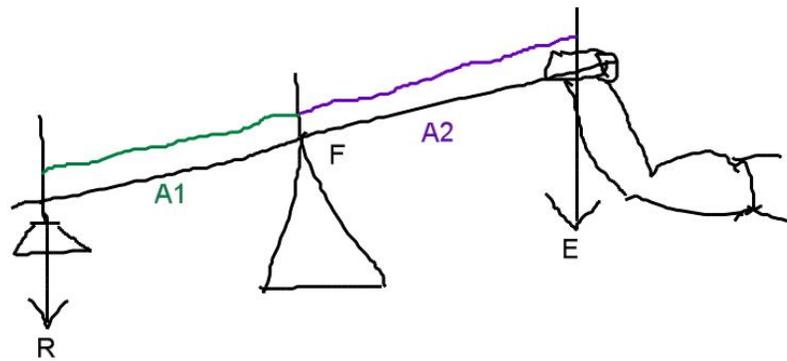


**Andrew Lovett**

# Modeling STEM Reasoning & Learning



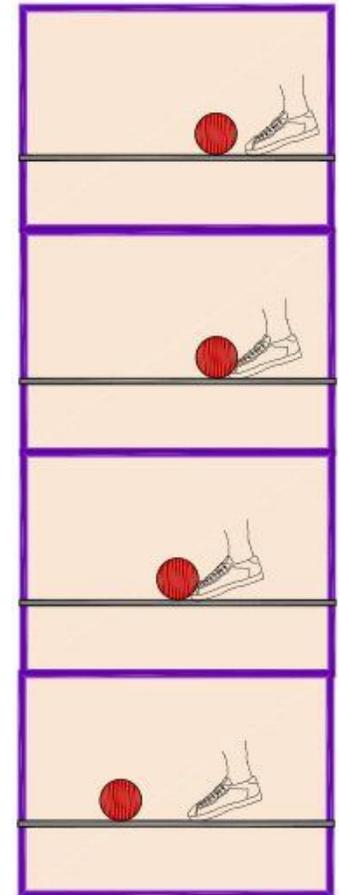
Solving ranking problems from conceptual physics textbook



Learning models of force:  
Trajectory of models similar to human students



Scott Friedman's Ph.D.



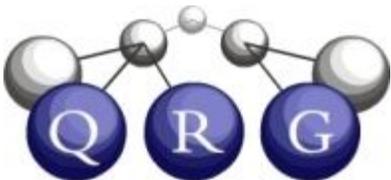
A lever has three basic parts. A fulcrum is a basic part of a lever.  
A force is a basic part of a lever. A weight is a basic part of a lever.  
(sketchForDiscourse "kb-resource://Figure1-1.sk" (DrsCaseFn DRS-3446218074-8197))  
F is the Fulcrum. E is the force. A2 is the distance between the weight and the fulcrum. A1 is the distance between the force and the fulcrum. A1 is an arm of the lever. A2 is an arm of the lever.

Kate Lockwood's Ph.D.: After reading simplified NL version of chapter, correctly answered 12/15 homework questions

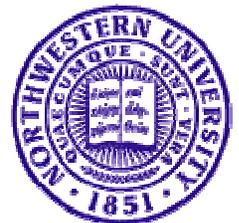


# Overview

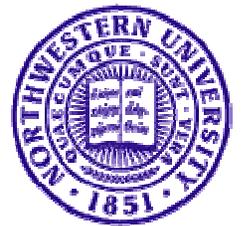
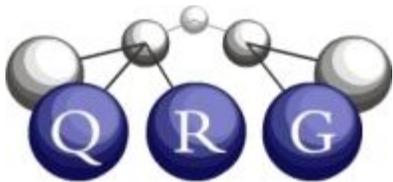
- Introduction to CogSketch ✓
- CogSketch Basics
- Visual processing in CogSketch
- CogSketch in Education
- Advanced features
  - Extending the KB, exporting knowledge...
- Wrap-up



**Your feedback will help us improve  
CogSketch**

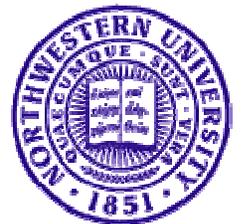
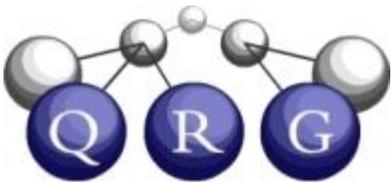


# CogSketch Basics



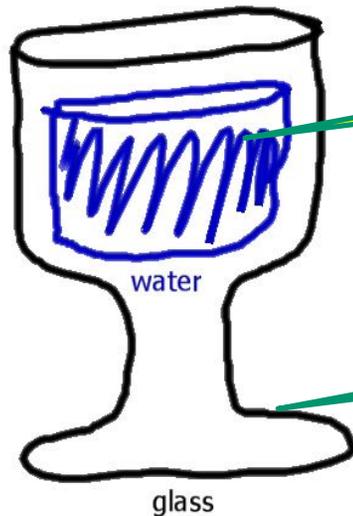
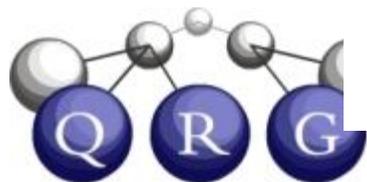
# This Section

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



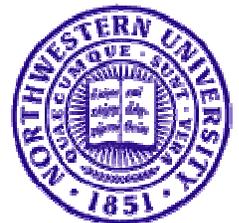
# Sketches are made of Glyphs

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

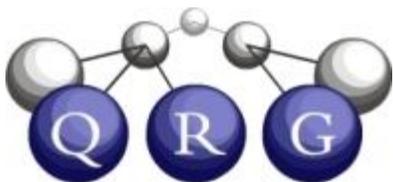
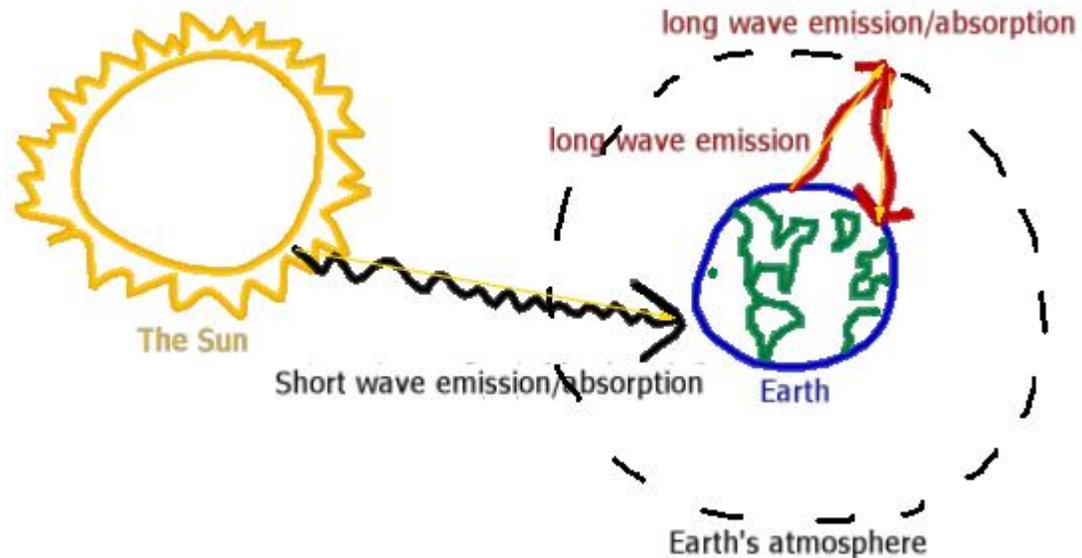
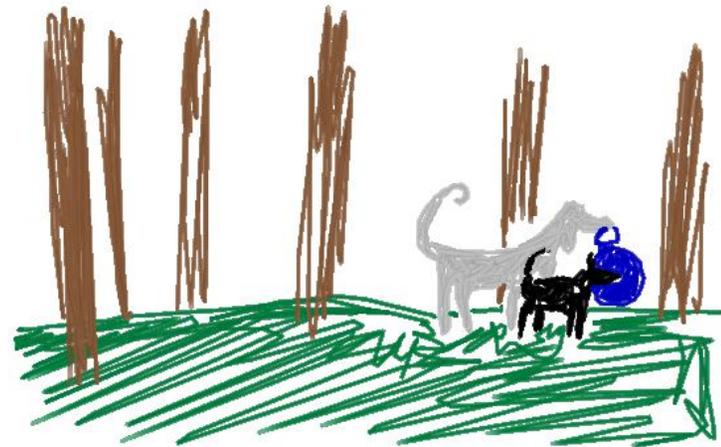
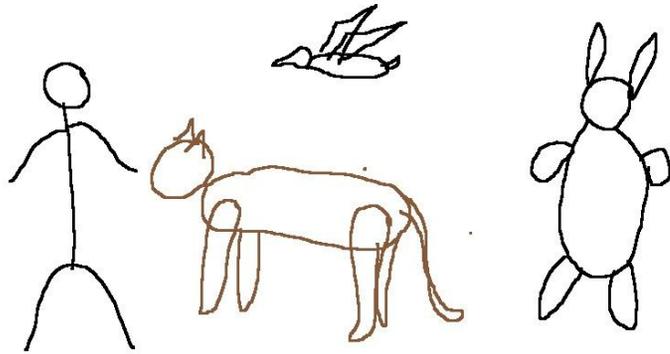


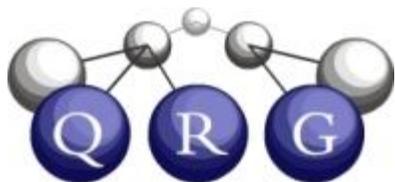
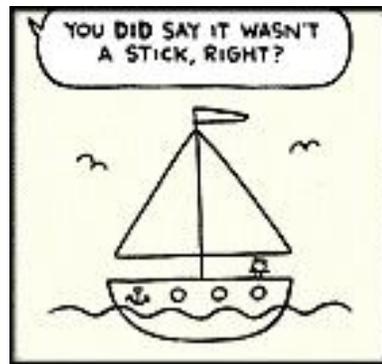
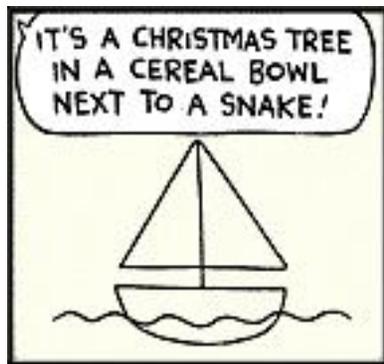
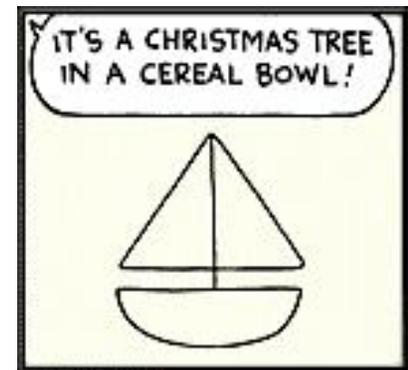
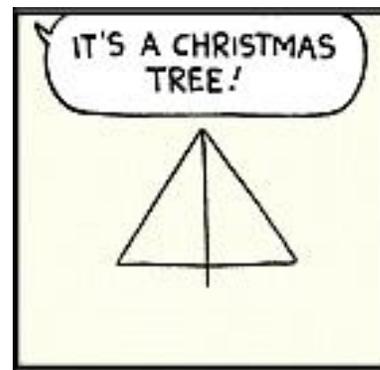
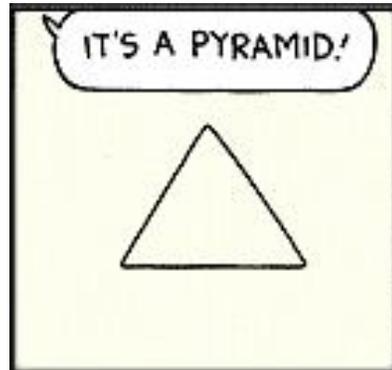
```
(nameString Object-12 "water")  
(isa Object-12 Water)  
...
```

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



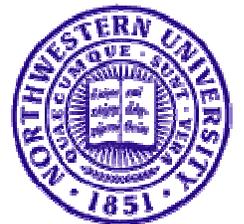
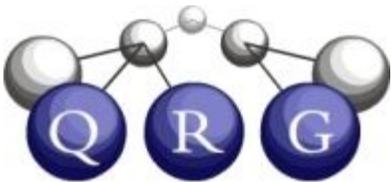
# Examples of Glyphs

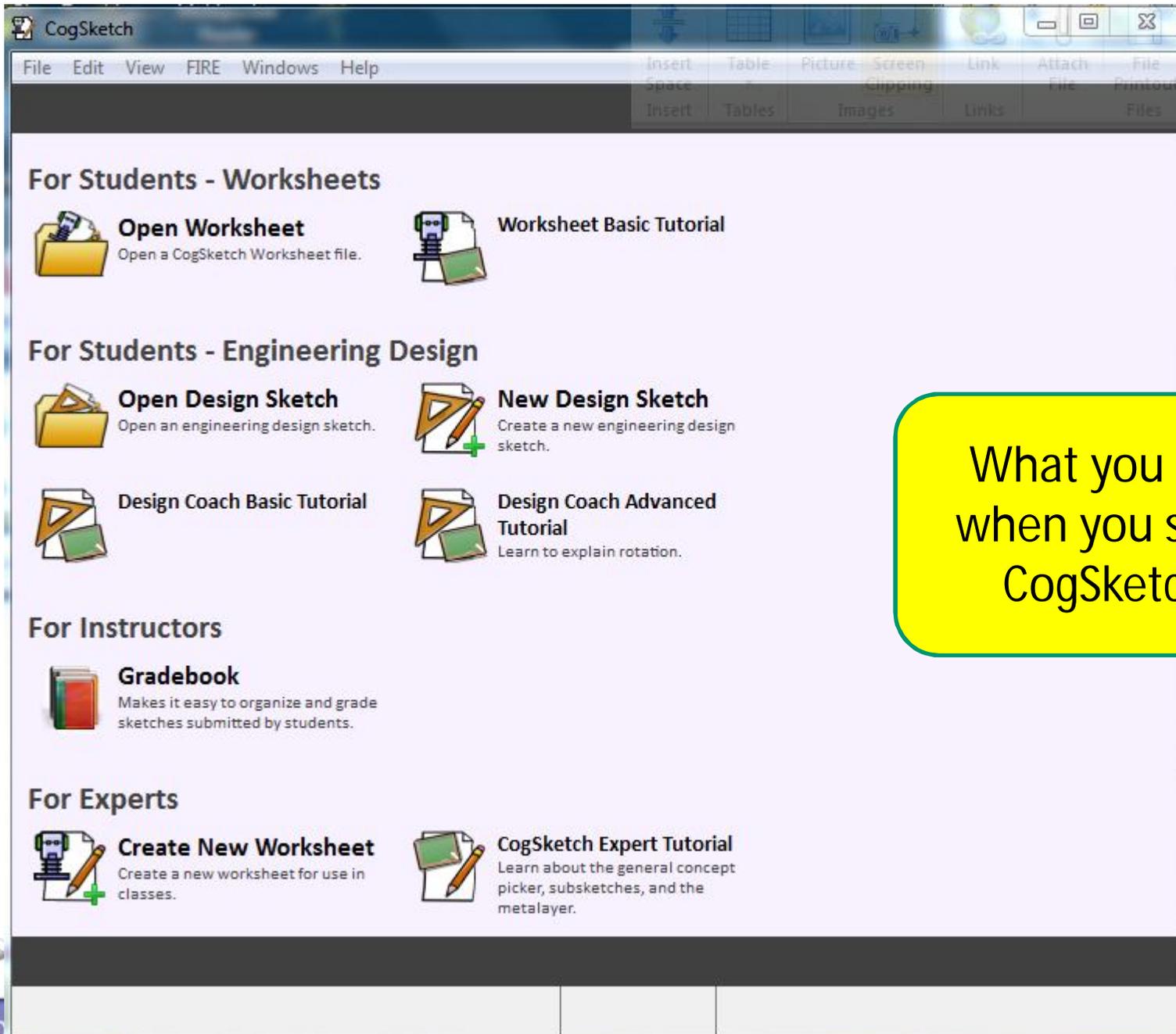




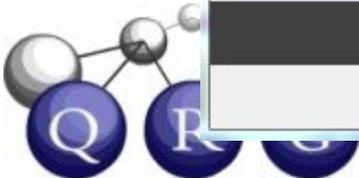
# Sketches have Structure

- 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



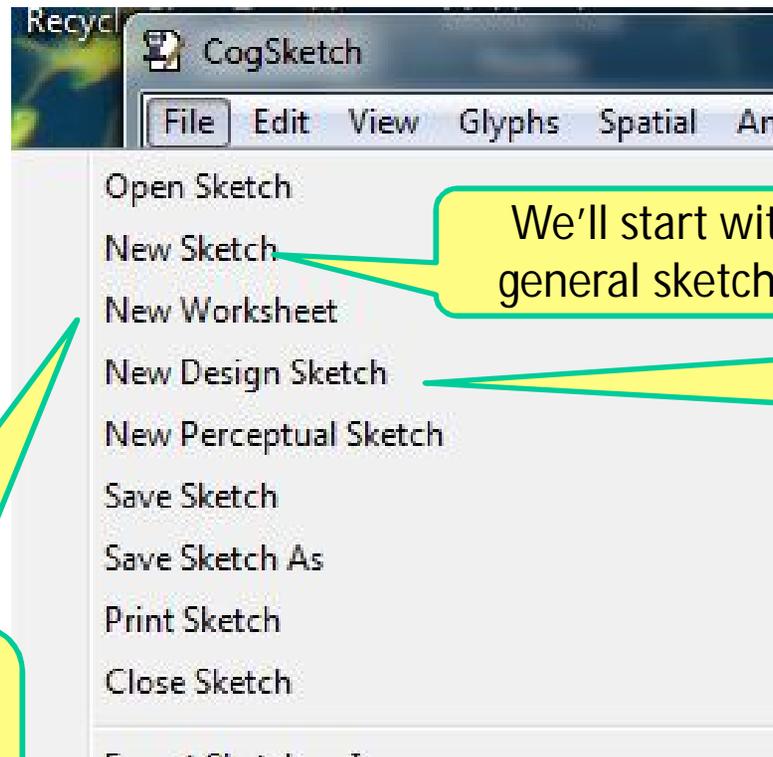


What you see when you start CogSketch



# Creating a New Sketch

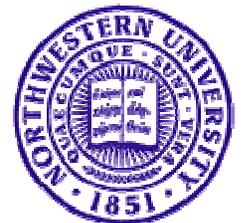
- There are four types of sketches



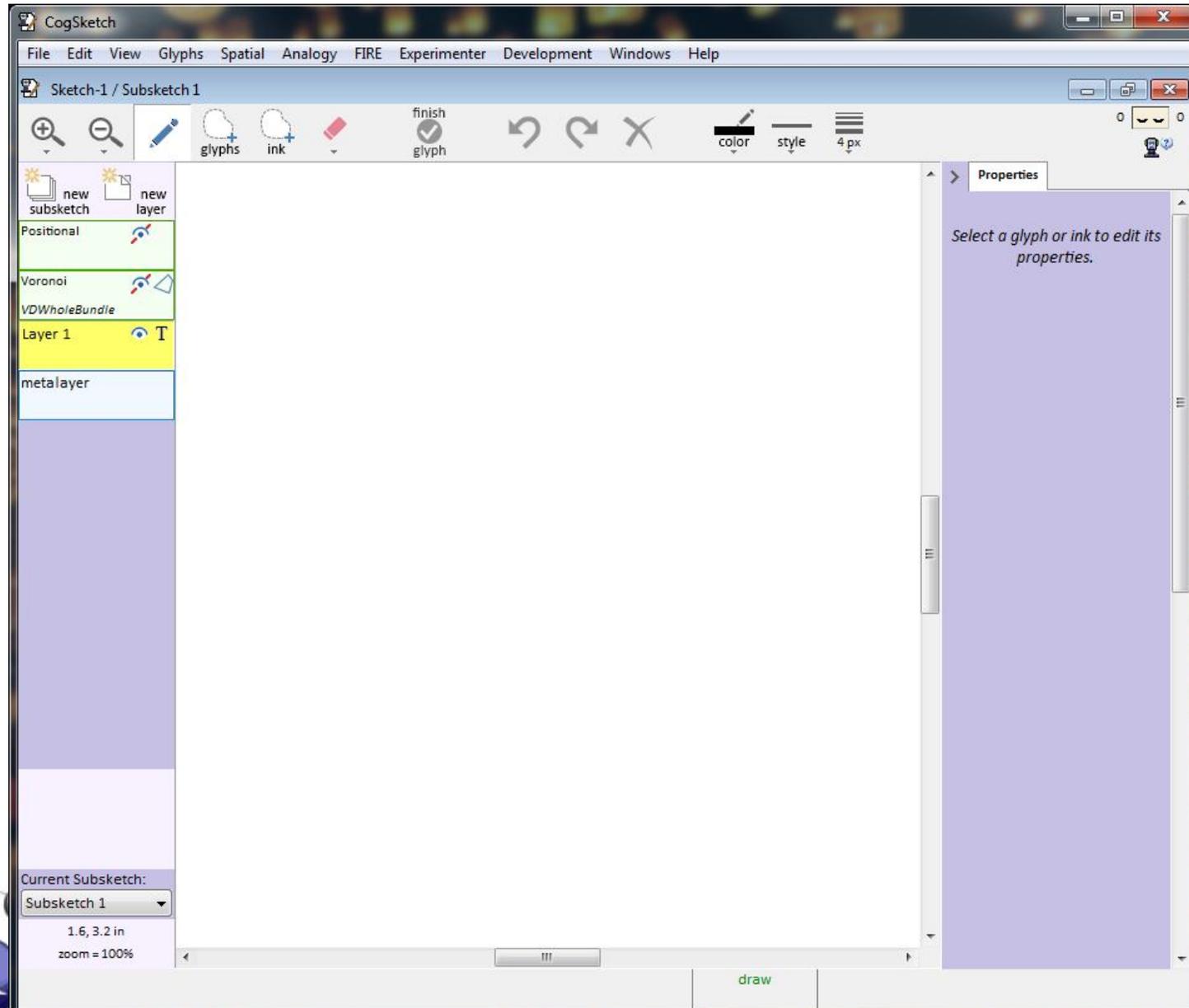
We'll start with general sketches

Perceptual Sketchpad and Design Coach will be discussed later

Worksheet authoring will be discussed later

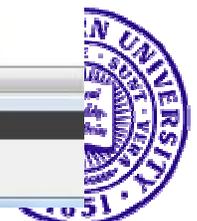
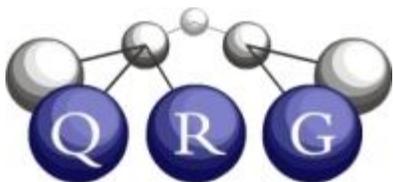
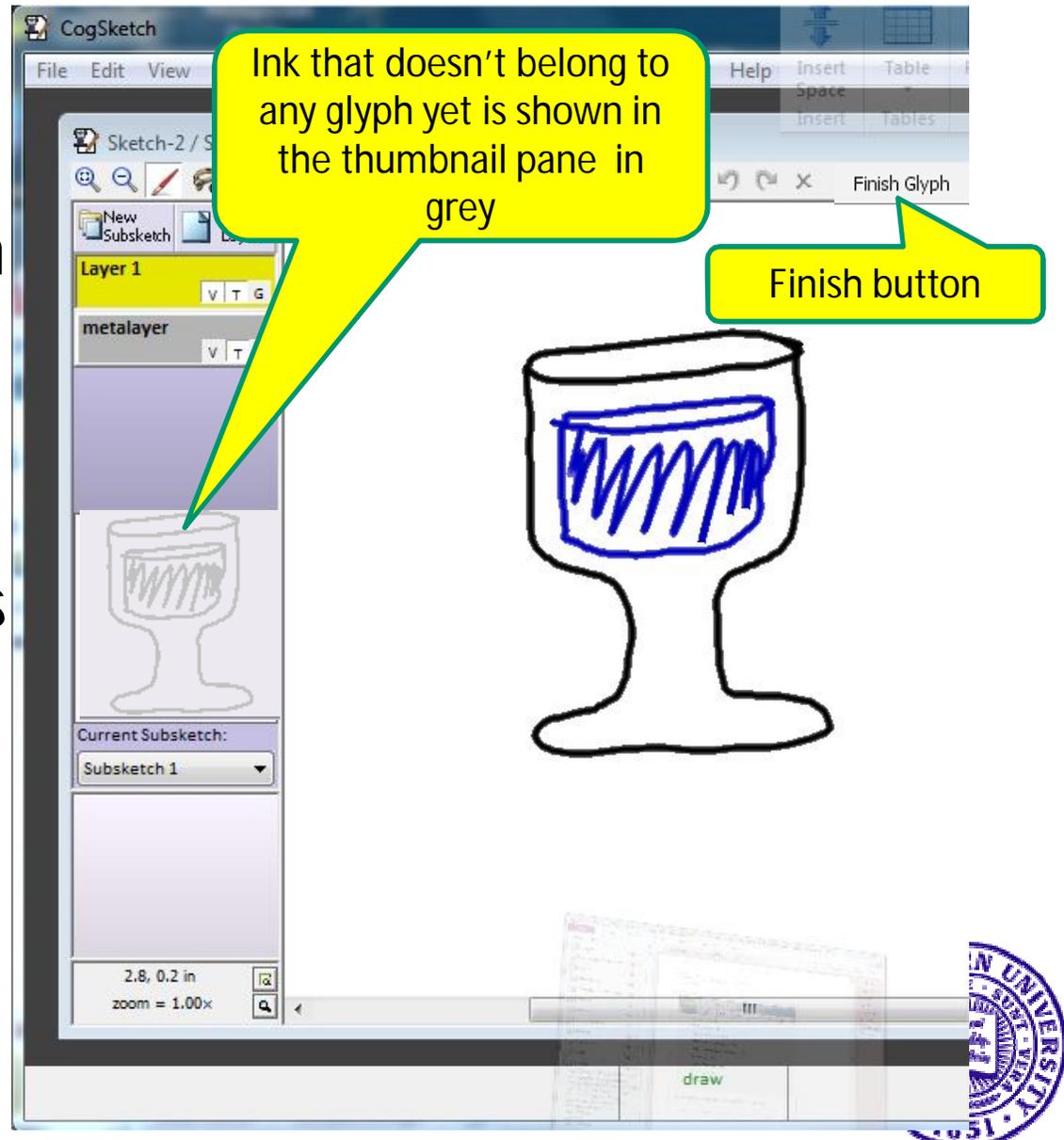


# What You Should See



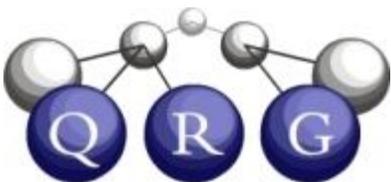
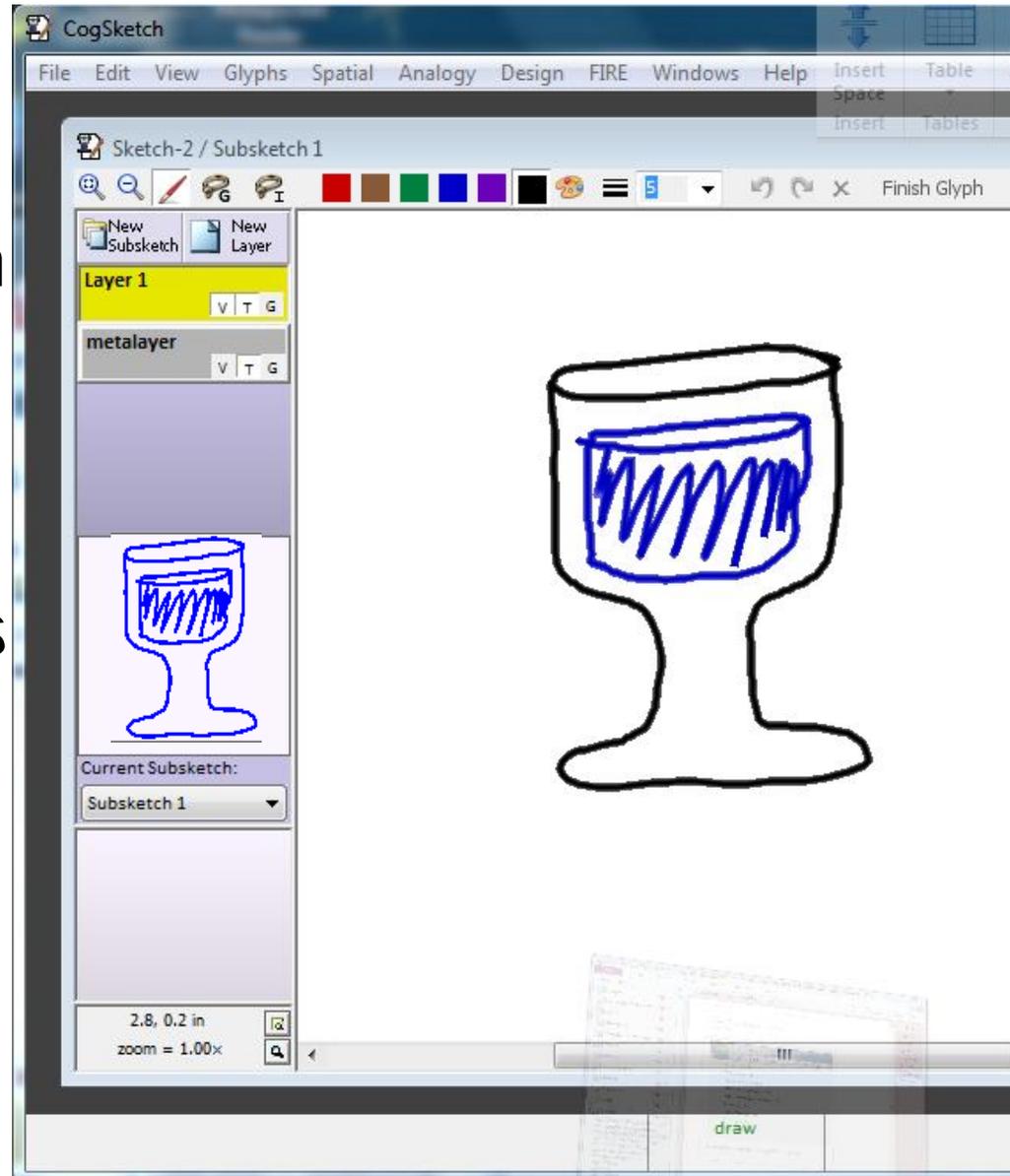
# Drawing a Glyph

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



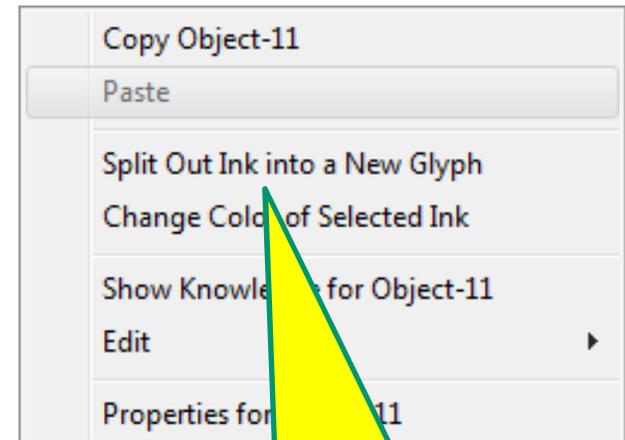
# Drawing a Glyph

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

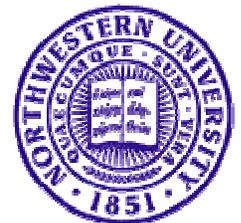
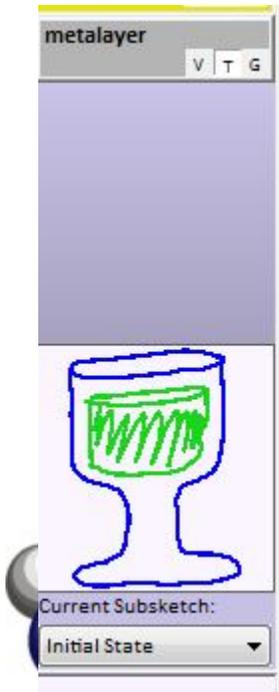


# Splitting ink into multiple glyphs

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

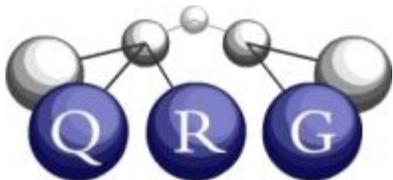


Right-click and choose split



# Merging Ink into Glyphs

Logo isn't part  
of the glass

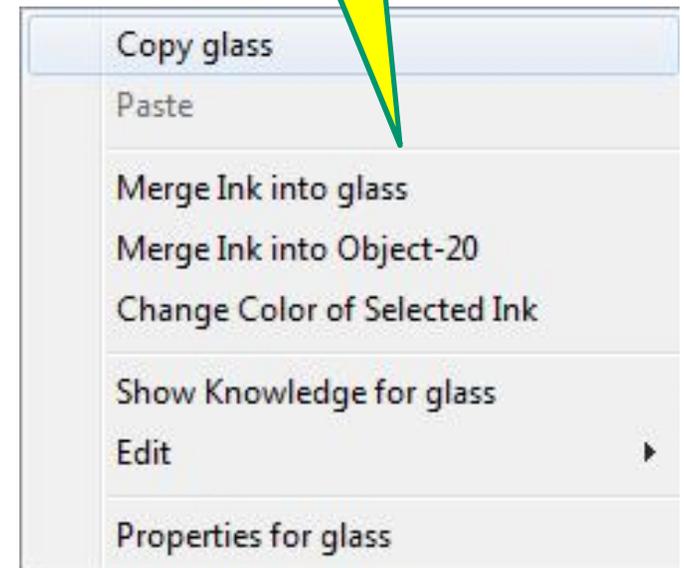


# Merging Ink into Glyphs

1. Select ink to be merged via the ink lasso

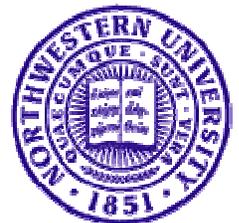
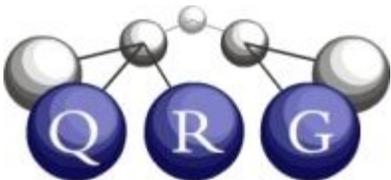


2. Select which is merged into the other



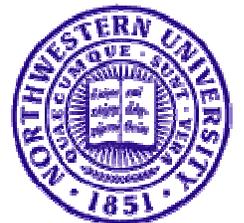
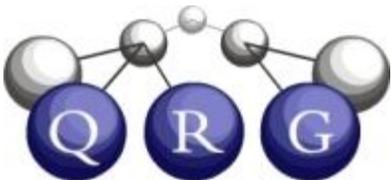
# Conceptual Labeling

- 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

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

1. Select the glyph to label using the Glyph lasso

3. Choose concept



# Choosing a Concept

Available Concepts:

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

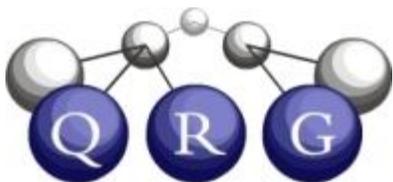
1. Start typing to see candidates

2. Comments help indicate which choice might be best

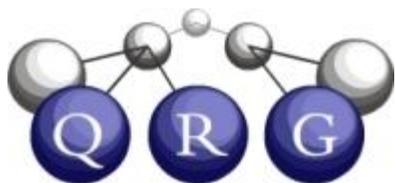
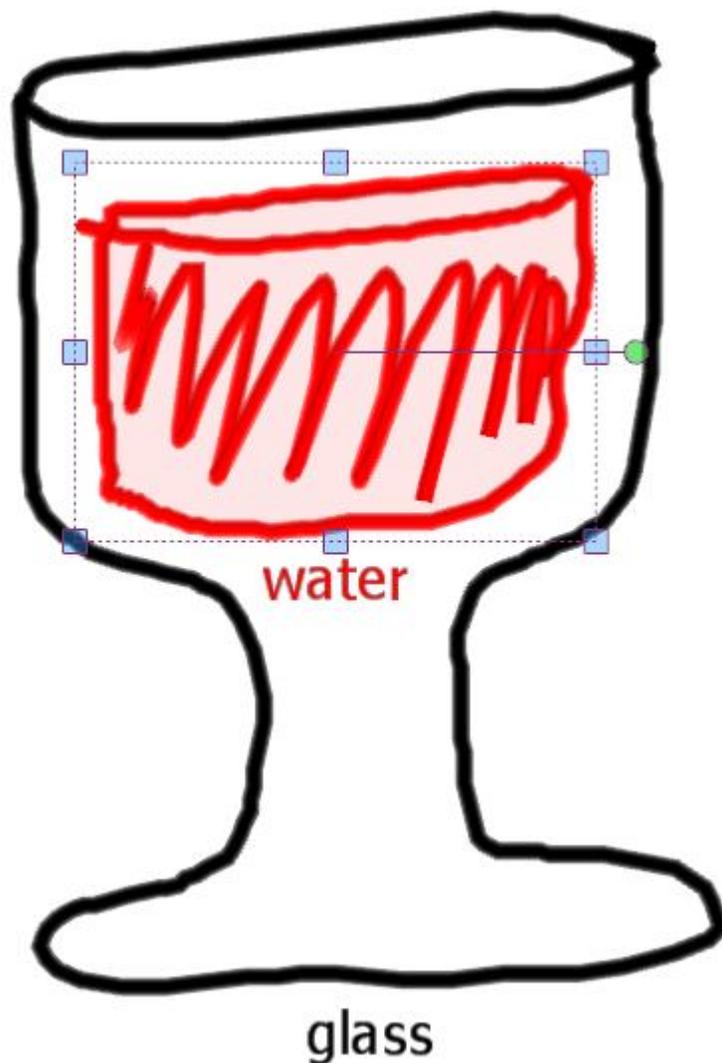
Available Concepts:

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

The collection of instances of Stemware made primarily of glass.



# Choosing a Concept, continued



Properties

Glyph Type: entity

Assigned Concepts:

- Water

Available Concepts:

Start typing in the box above to search the available collections.

An instance of ChemicalCompoundTypeByChemic 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 ...

Name:

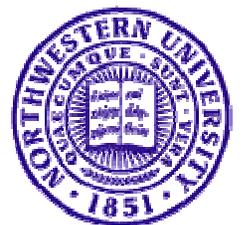
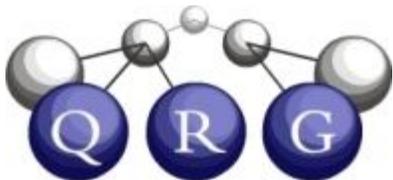
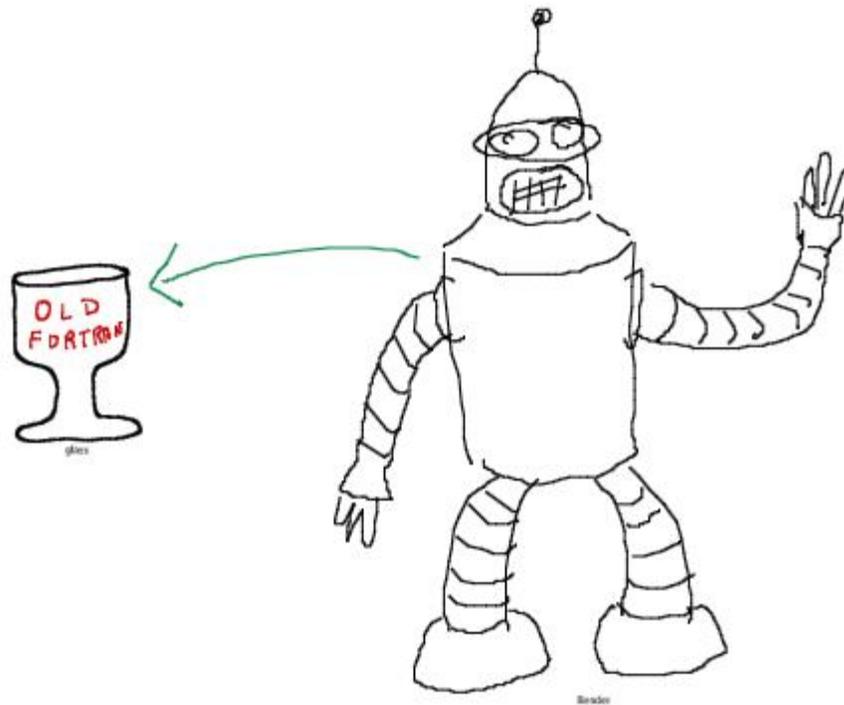
water

Ink Properties

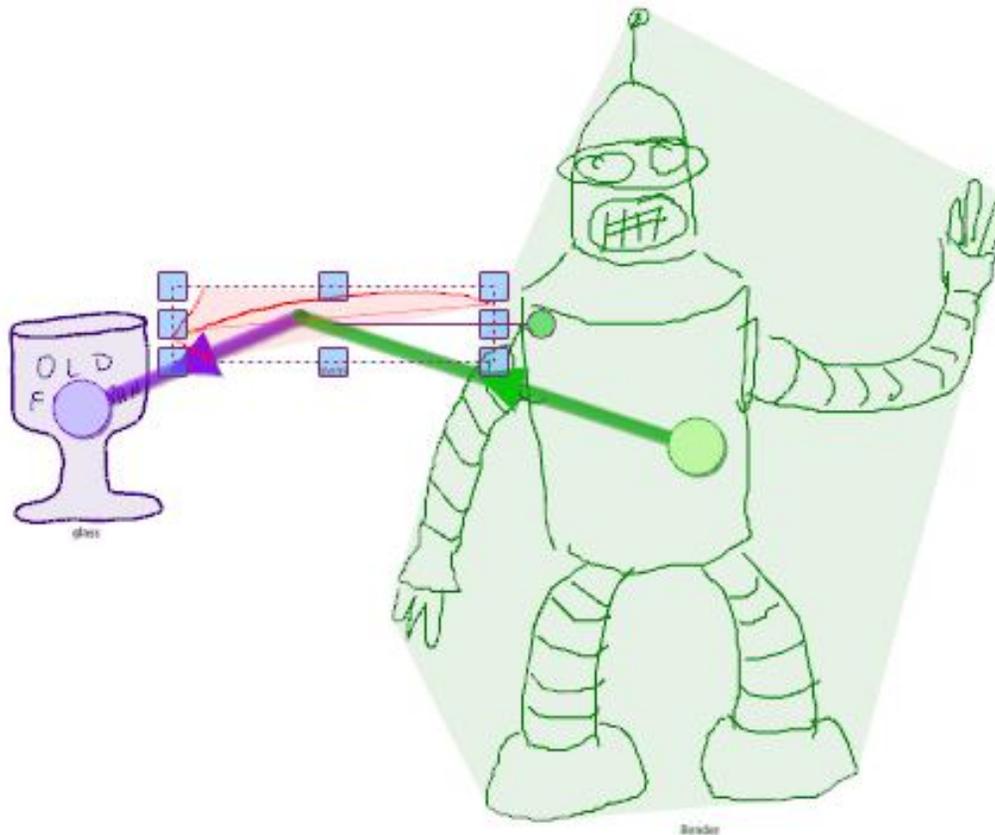
color style 5 px

# Relation Glyphs

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



# Labeling a Relation Glyph



Properties

Glyph Type: relation

What is this?

owns

*Start typing in the box above to search the available relations.*

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

Bender property glass.

Name:

owns

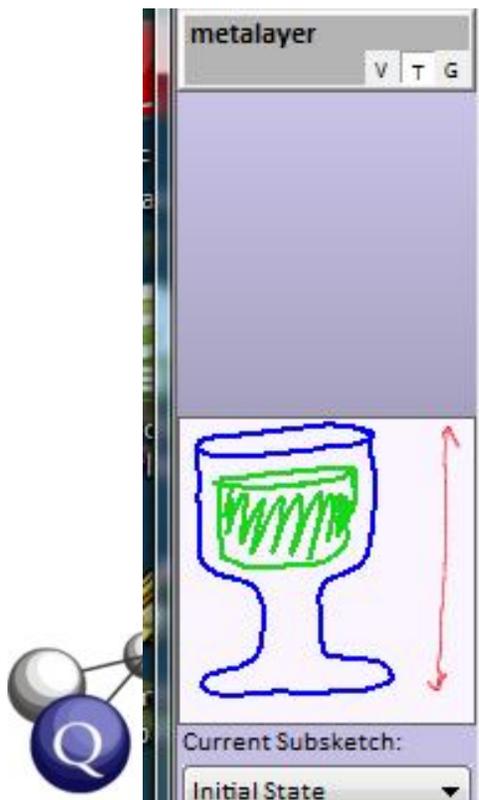
Ink Properties

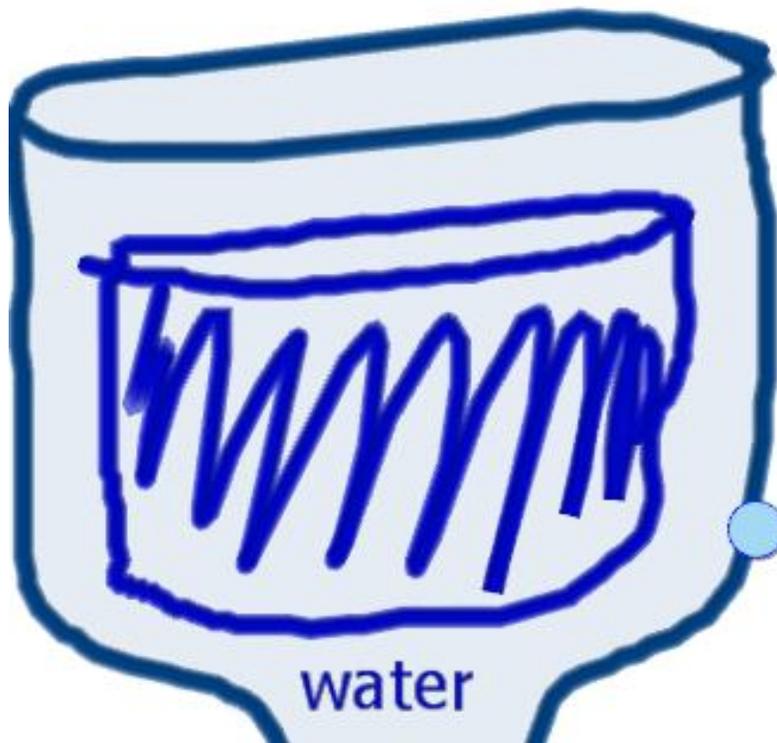
color style 3 px



# 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



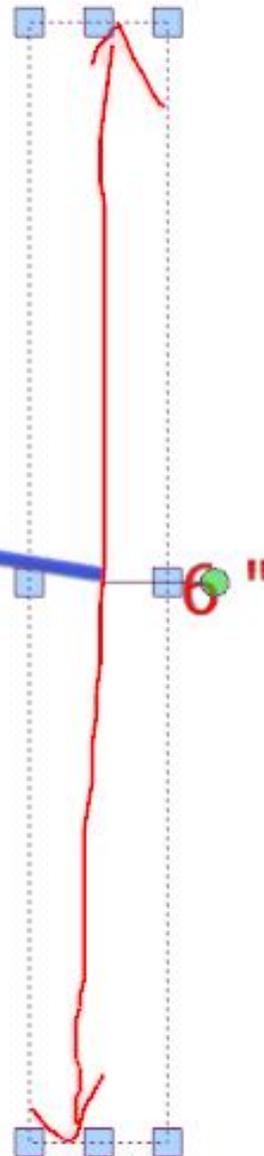


water

# Labeling an Annotation Glyph



glass



Properties

Glyph Type:

What is this?  
 LengthIndicator

Value:  
 6 inches

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

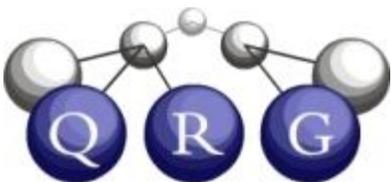
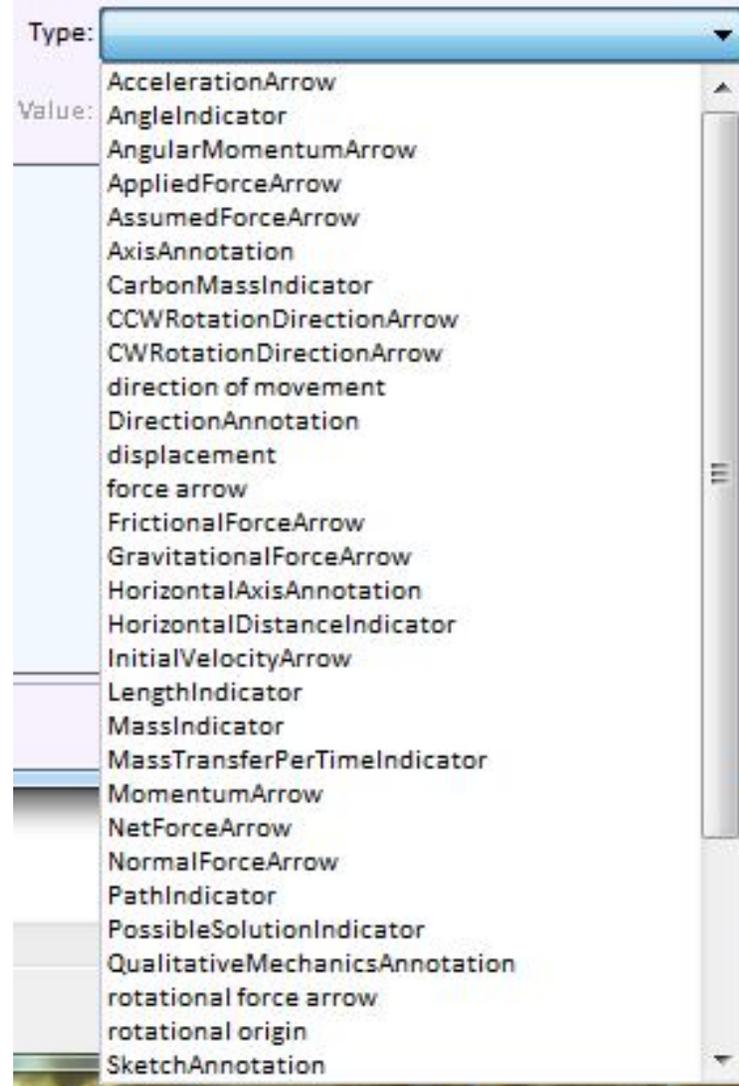
Name:

Lengthindicator applies to:  
• glass

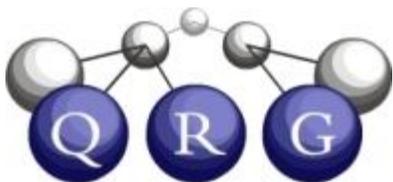
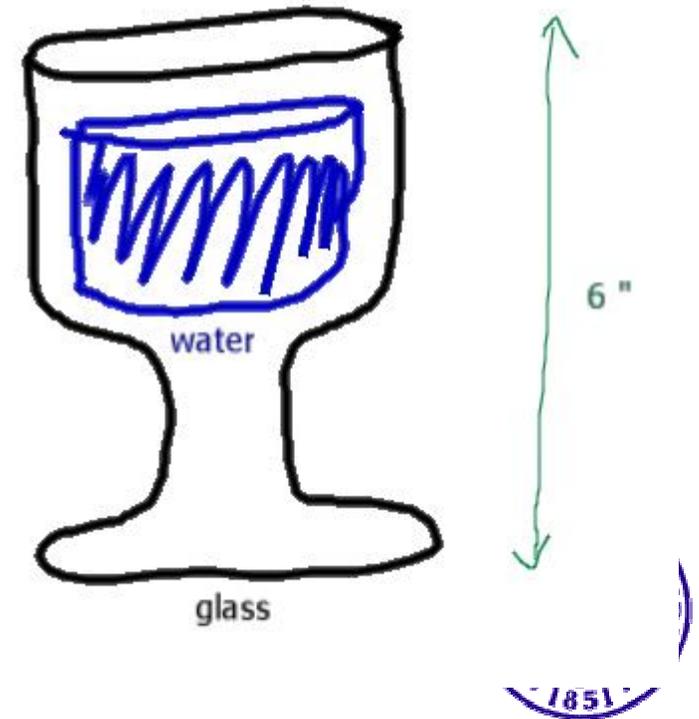
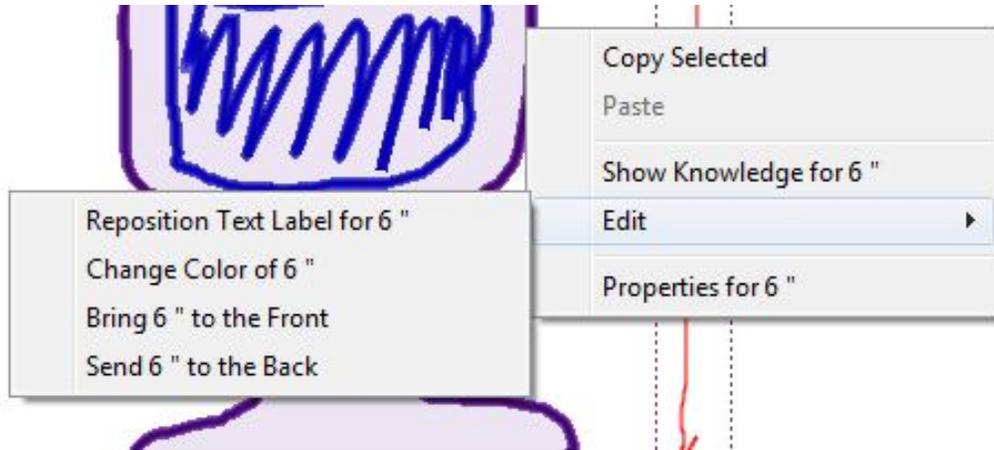
Ink Properties  
color style 1 px



# Types of Annotations

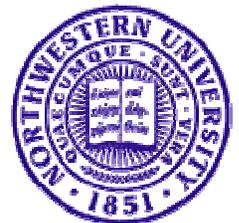
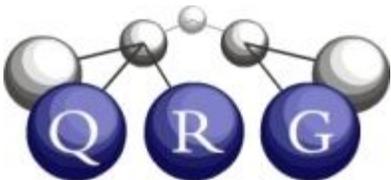


# Neatening your Sketch



# What CogSketch Does with This Information

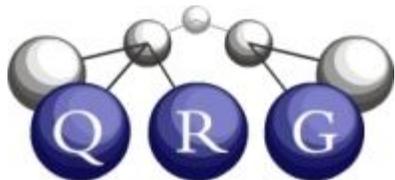
- It enables CogSketch to reason about the objects you sketched
  - e.g. simple qualitative mechanics is built-in
  - Used in Design Coach, mentioned later
  - You can hook up your own reasoners to it
- It enables CogSketch to match sketches
  - e.g., sketch worksheets for education compare a student's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



# Status Indicators



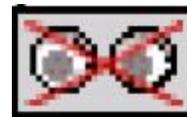
Waiting. Happens while you are drawing/moving something, postponing visual processing until you are finished, for responsiveness.



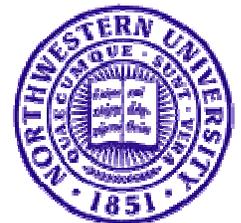
Idle. Digits indicate the number of tasks queued



Both processors running



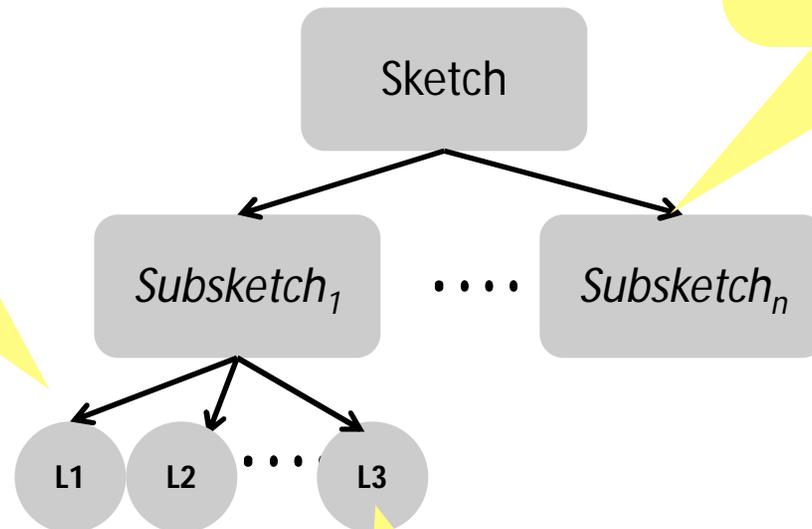
Crashed. (Very rare) Touch to restart



# The Structure of Sketches

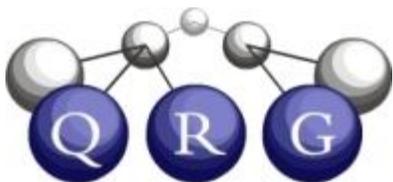
Multiple layers in the same subsketch may be visible at the same time, like in drawing programs or acetate sheets over paper.

Each sketch must have at least one subsketch, and each subsketch must have at least one layer



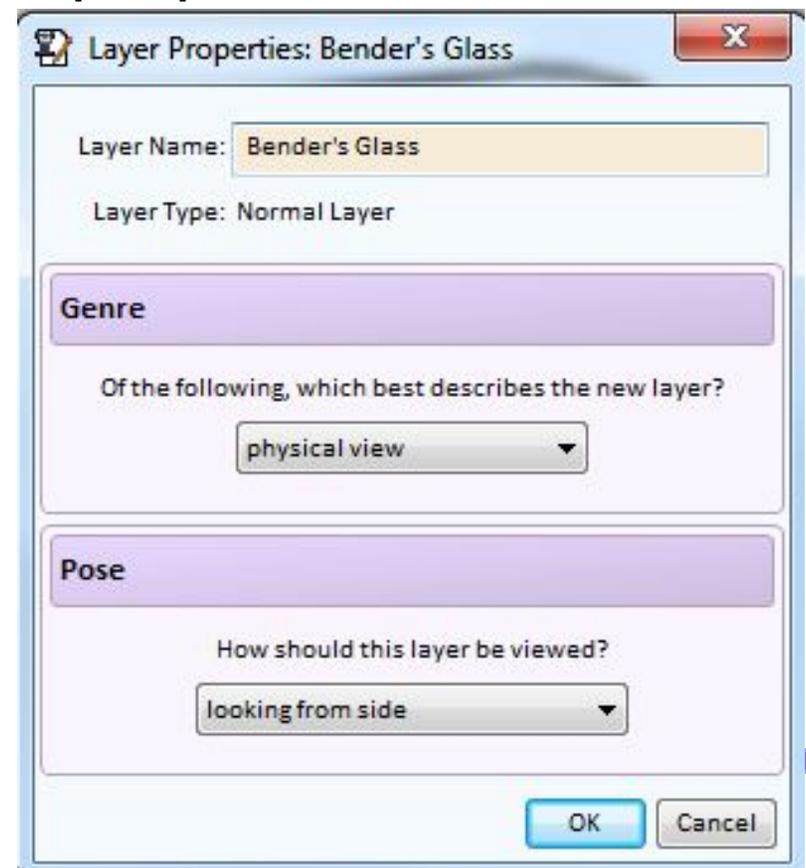
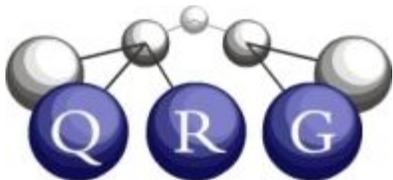
Only one subsketch is active at one time

By default, spatial relations are only computed between glyphs on the same layer



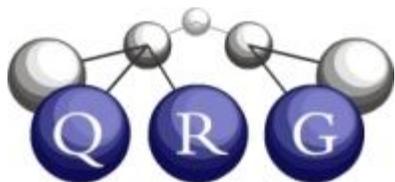
# How Layers are Interpreted

- 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

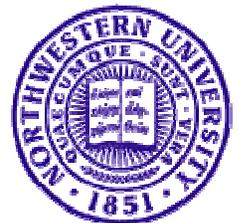
- *Abstract*: Visual relationships between glyphs provide no information about spatial relationships between them
- *Discrete graph*: Visual contact relationships important, but other visual properties (e.g. distances and locations) are not
  - Example: Concept maps
- *Geospatial*: Visual coordinates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



Spatial coordinates  
(unless genre is  
abstract or  
discrete graph)

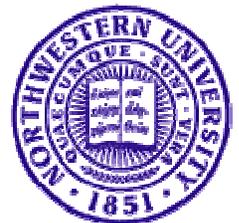
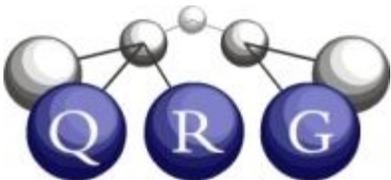


Visual coordinates

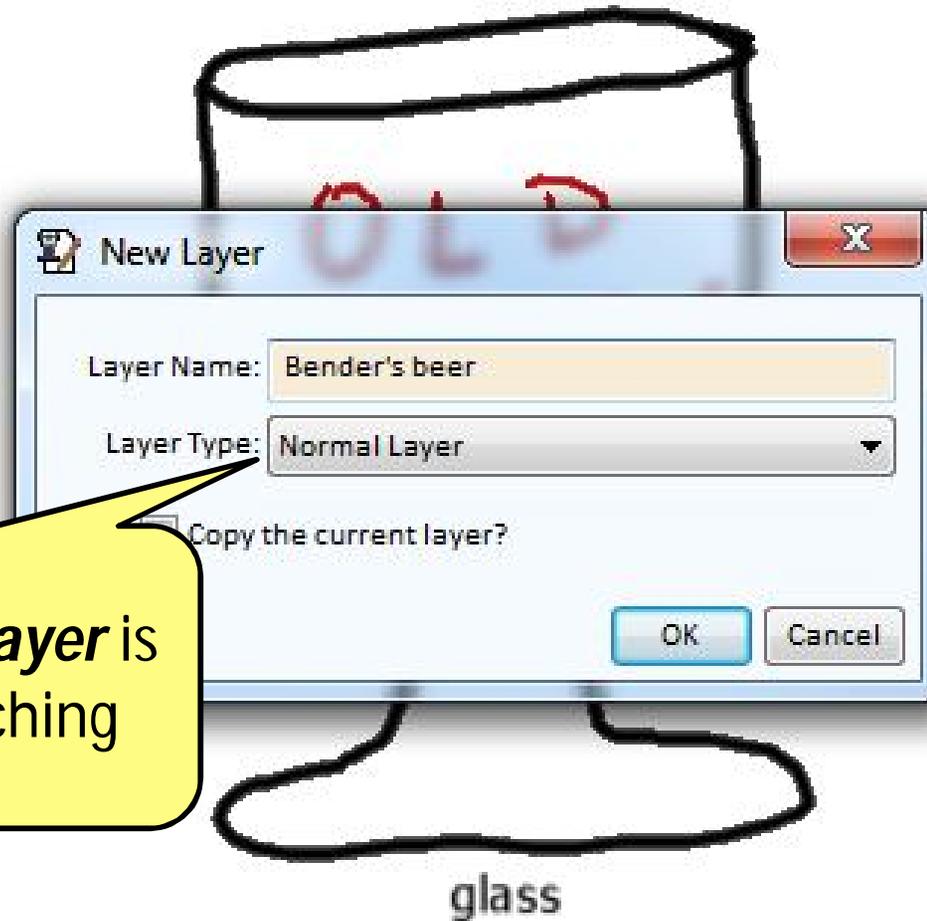
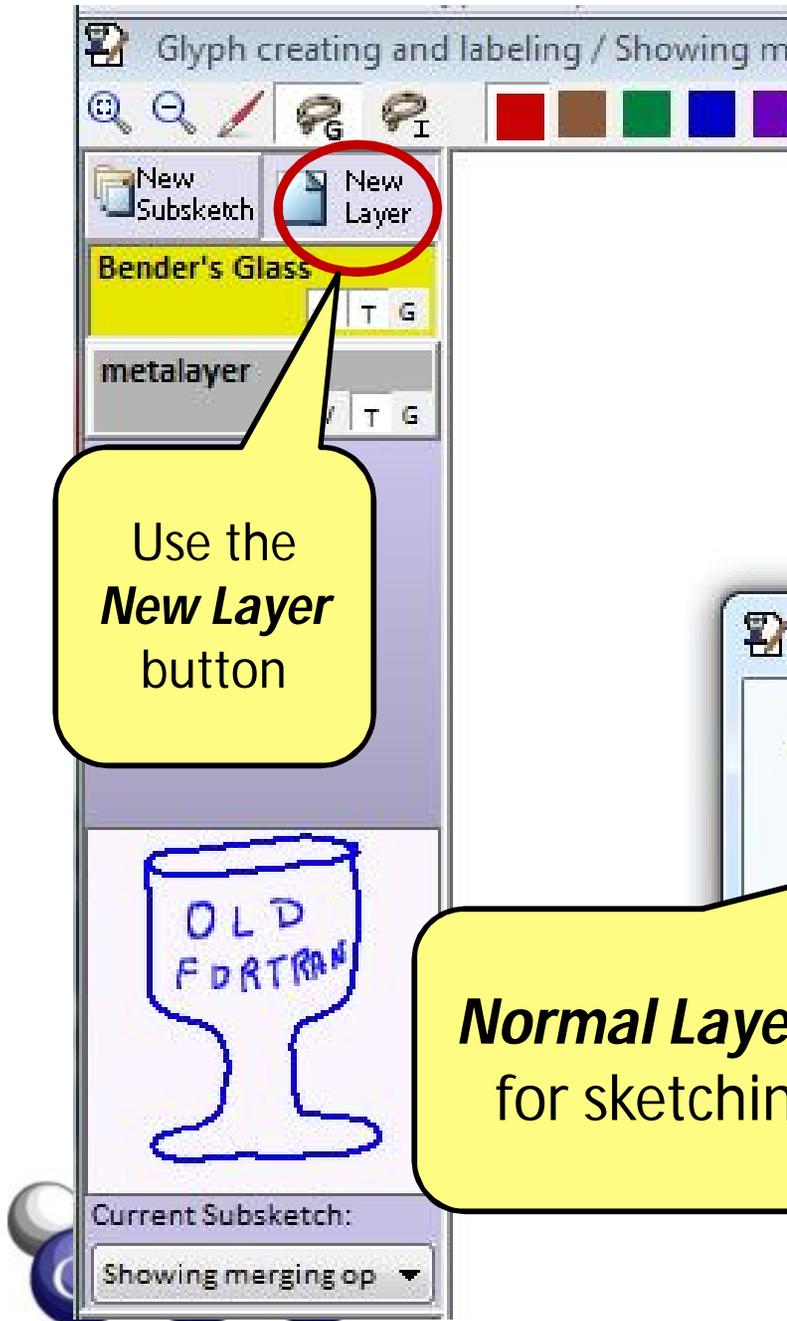


# Pose

- *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



Positional	
Voronoi	
VDWholeBundle	
Bender's beer	<del>T</del>
Bender's Glass	<del>T</del>
metalayer	

~~T~~ text visible?  
 layer ghosted?

# Controlling Visibility



glass



glass

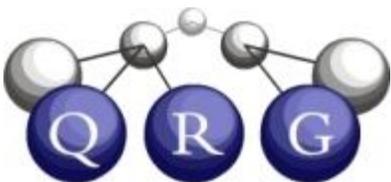
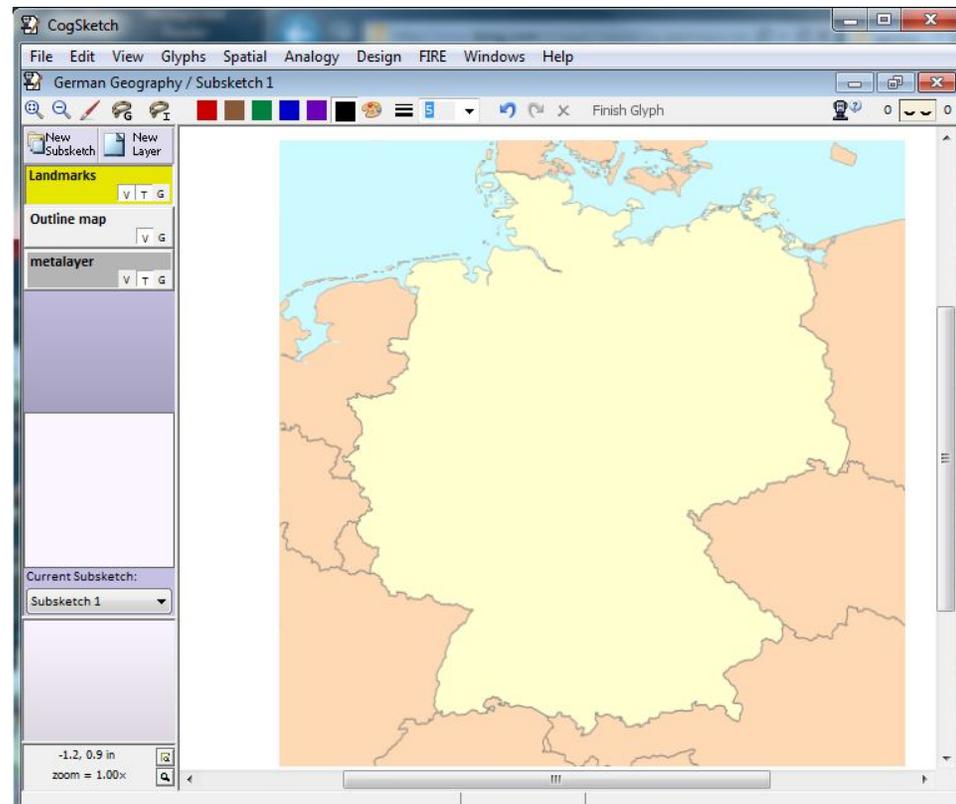
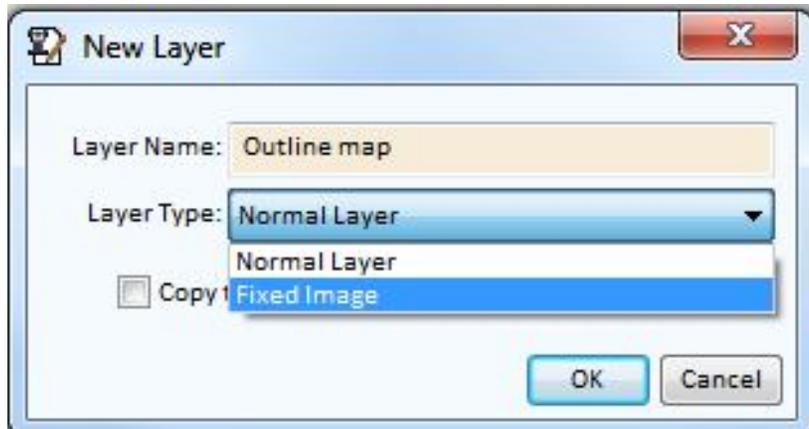


glass



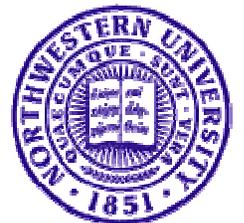
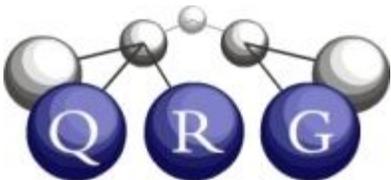
# Adding a Bitmap Layer

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

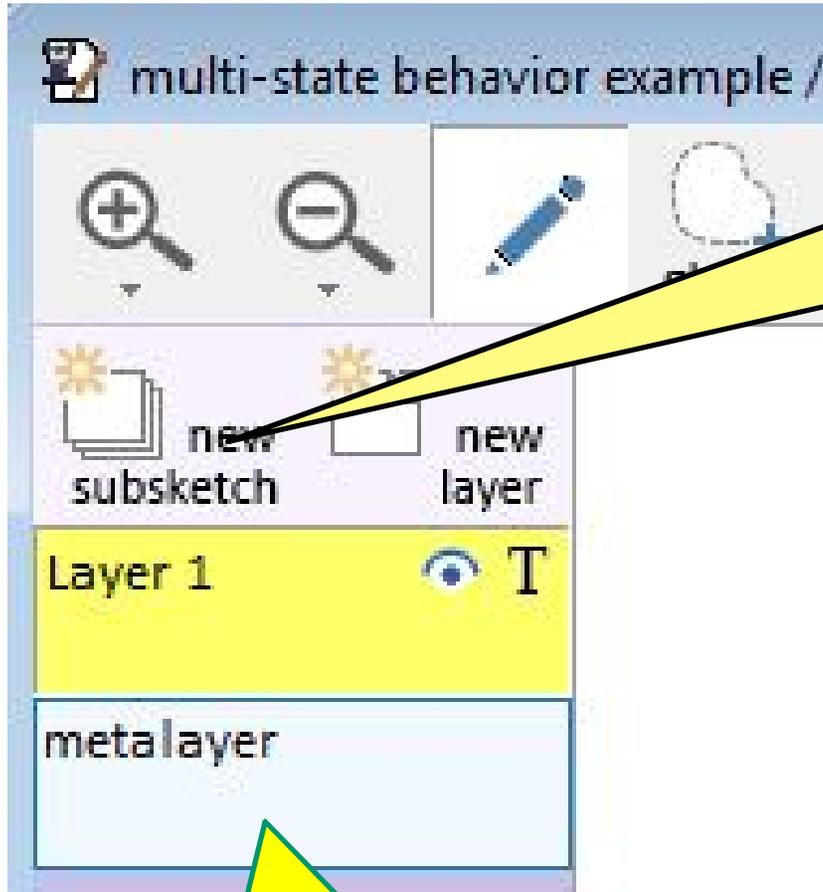


# Uses for Multiple Subsketches

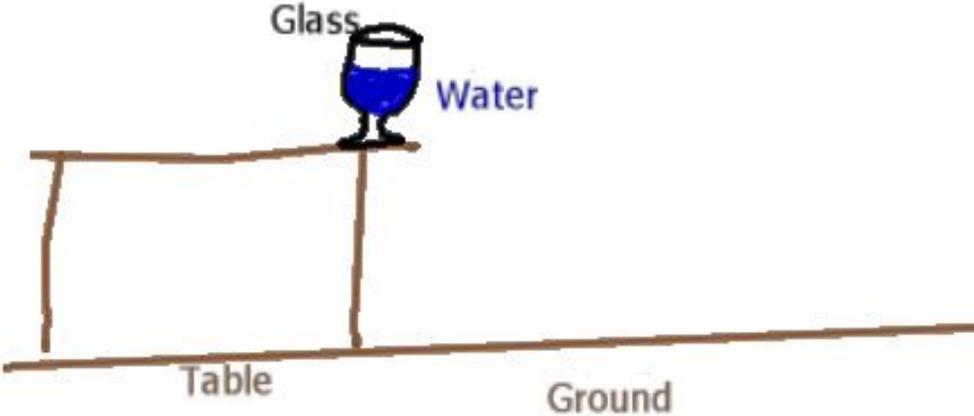
- Describing a complex behavior
  - Each subsketch might represent a distinct qualitative state
  - Can create *comic graphs*, a generalization of comic strips, that allow branches and joins in addition to sequentiality
- Describing alternatives
- Describing something from multiple perspectives



# Adding a Subsketch



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

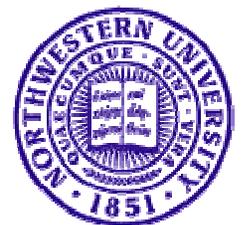
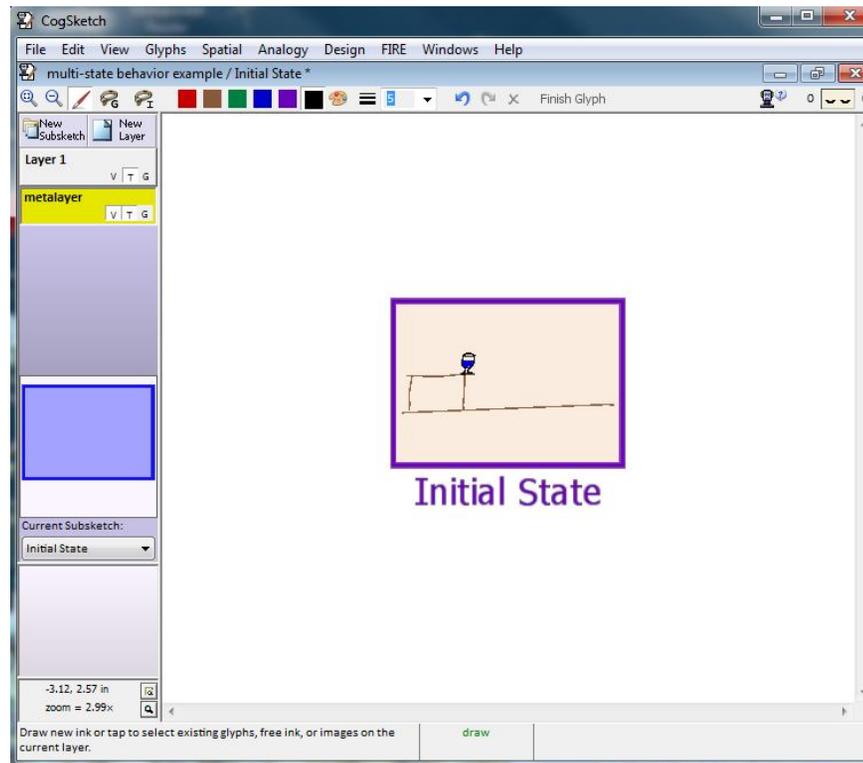
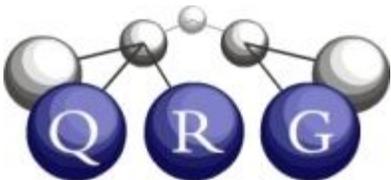


The *metallayer* lets you manipulate subsketches



# The Metalayer

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



# Conceptually Labeling a Subsketch

Subsketch Name:

**What does the subsketch represent?**

StaticSituation

Event  
StaticSituation  
PartiallyTangible

More Choices

Subsketches are used to represent something else -- for example, a subsketch can be used to represent an event or a physical object. Select the things your subsketch represents in the yellow box to the right, and use the arrow button above to add them to the list of things represented by this subsketch (the white box above).

Things represented by the subsketch can be removed using the 'X' button.

Pushing the 'More Choices' button will give you a wider range of choices for the types of things your subsketch can represent.

OK Cancel

Default = a static configuration



# Conceptually Labeling a Subsketch

Subsketch Name: Initial State

What does the subsketch represent?

StaticSituation

PhysicalSituation

PhysicalSchema  
PhysicalSeries  
PhysicalSituation  
PhysicalSkillTesting  
PhysicalStuffAndObjects-Topic  
PhysicalSynthesis

Select the things your subsketch represents by typing the name of a collection in the yellow box at the top right. The larger yellow box to the right will show valid completions; use the arrow button to add one of the completions the list of things represented by this subsketch.

A specialization of Situation-Localized. Each instance of PhysicalSituation is a spatially localized situation involving one or more physical objects or stuffs (see situationConstituents). Important specializations include PhysicalEvent and AilmentCondition.

Fewer Choices

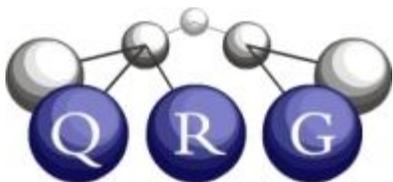
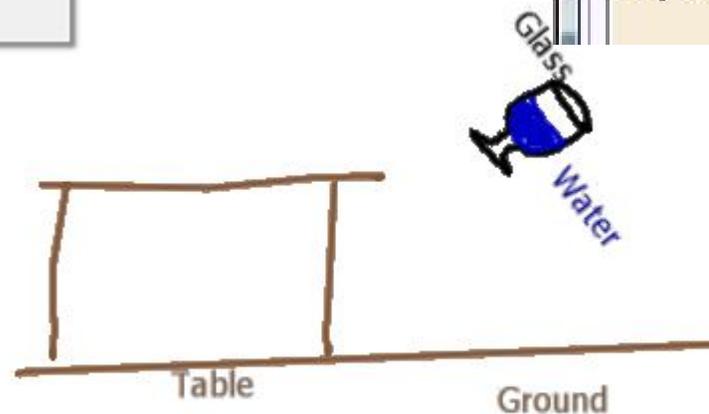
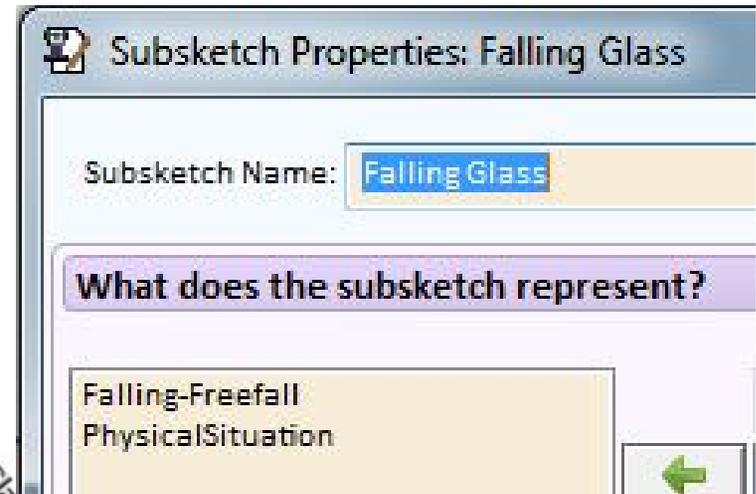
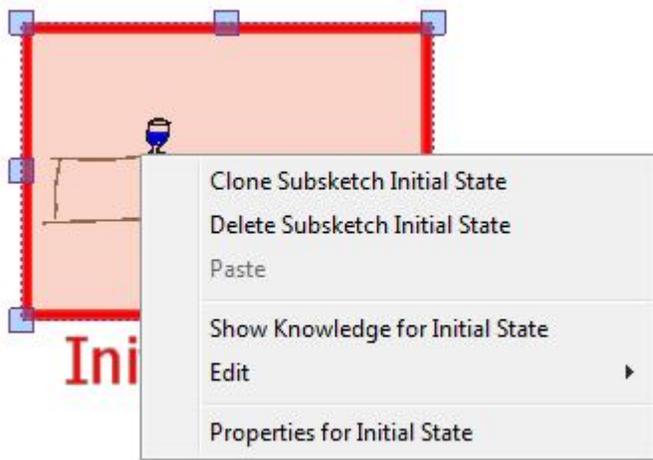
OK Cancel

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

The diagram illustrates the process of linking two behaviors. The top part shows two separate states: 'Initial State' (a glass on a table) and 'Falling Glass' (a glass falling from the table). A yellow arrow points from the Initial State to the Falling Glass, with the word 'after' written below it. The bottom part shows the same two states, but they are now linked. A red arrow points from the Initial State to the Falling Glass, and a green arrow points from the Falling Glass back to the Initial State. The word 'after' is written in red below the red arrow. The Initial State is now a purple box, and the Falling Glass is a green box. A properties panel on the right shows the 'after' relation selected.

**Initial State** after **Falling Glass**

**Initial State** after **Falling Glass**

**Properties**

Glyph Type: relation

What is this?

after

Start typing in the box above to search the available relations.

A PrimitiveTemporalPredicate that relates two points in time. (after LATER EARLIER) means TimePoint LATER is after (occurs later in time than) TimePoint EARLIER.

**Falling Glass after Initial State.**

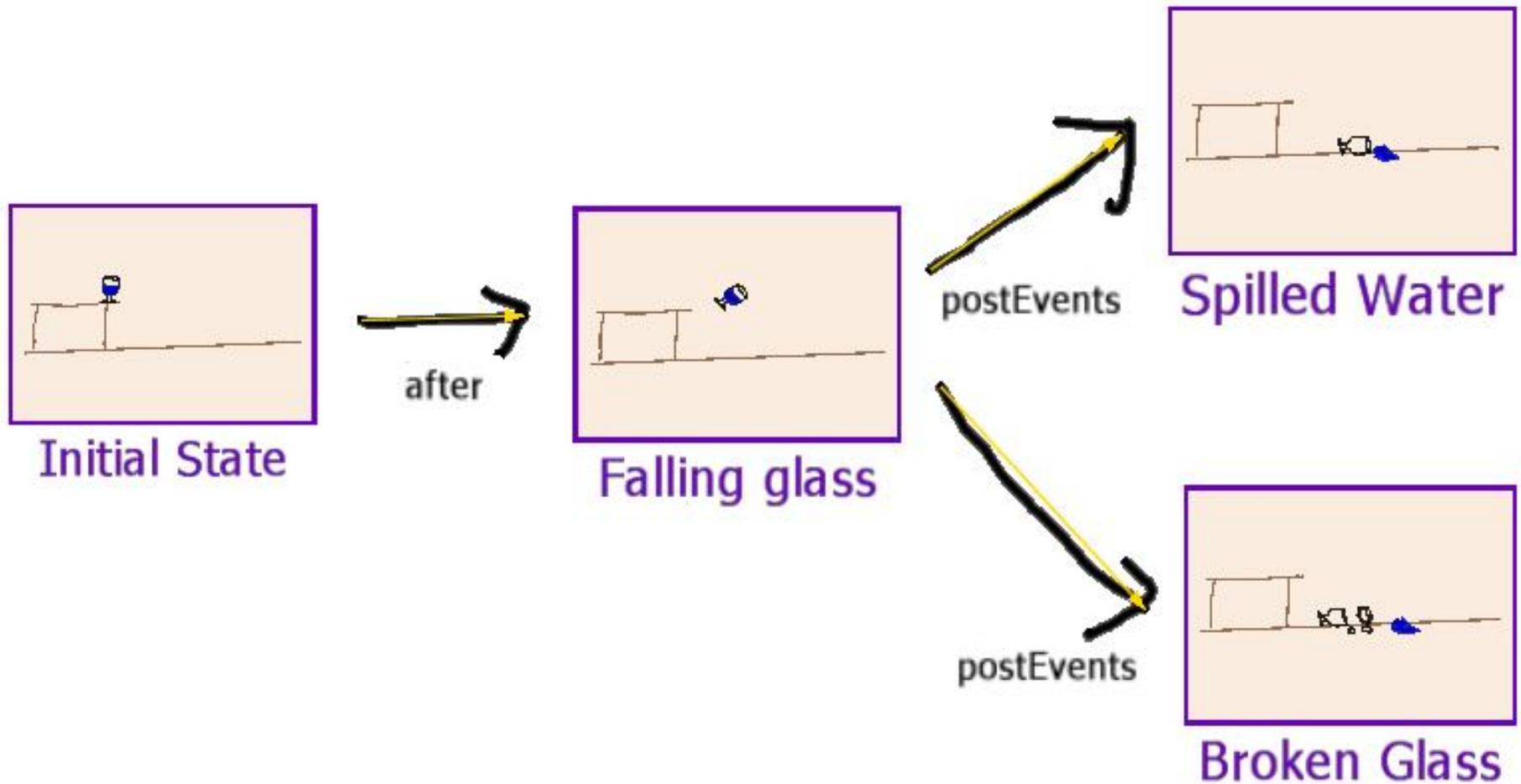
Name:

after

Ink Properties

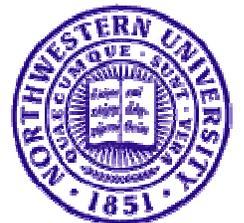
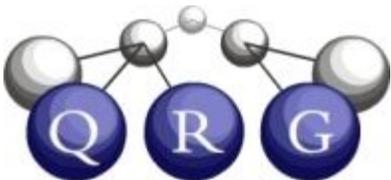
color style 4 px

# A Comic Graph

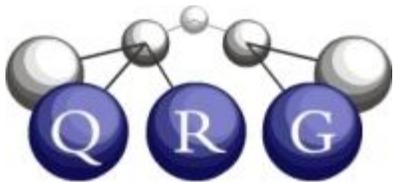


# What You Have Seen

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



# Visual Processing in CogSketch



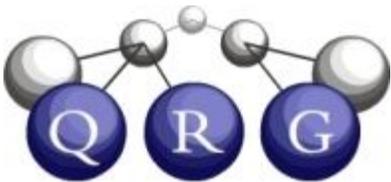
# Some Preliminaries

Visual versus Spatial relationships:

- Visual relationships: Computed over glyphs
- Spatial relationships: Hold between what is denoted by the glyphs
- Visual relationships + genre + pose  $\rightarrow$  Spatial relationships

Our visual computations are inspired by psychological evidence when available

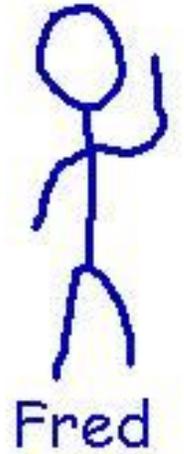
- Best guesses otherwise
- We expect it to continue to evolve



# Glyphs

Glyphs have two parts: *Ink* and *Content*

- Content = the entity represented by the glyph
  - Instance of some collection in the KB
- Ink = visual representation of the content
  - Consists of all of the ink drawn between button presses



Visual properties are computed on the ink

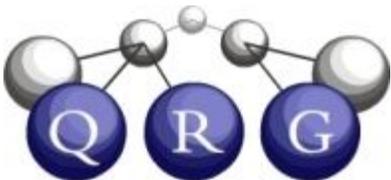
- Only coarse visual properties computed automatically
  - Bounding box
  - Closed contour (ink needn't be connected)
  - Major/minor axes
- Small set of visual relationships between glyphs
- Decompositions, other visual relationships computed on demand
  - See CogSketch\_Spatial\_Reasoning.pdf for API



# Qualitative Spatial Reasoning

Claim: Symbolic vocabularies of shape and space are central to human visual thinking (cf. Forbus 1980; Forbus, Ferguson & Usher 2001; Kosslyn et al., 1989)

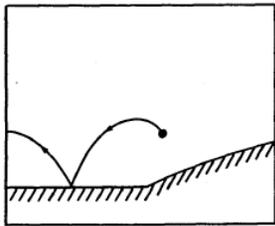
- They are computed by our visual system
- Their organization reflects task-specific conceptual distinctions and conventional symbol systems as well as visual distinctions
- They provide the bridge between conceptual and visual representations



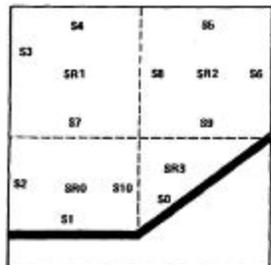
# Metric Diagram/Place Vocabulary model

Metric Diagram: Quantitative, visual representations and processing

Place Vocabulary: Task-specific qualitative representations of shape and space, grounded in the metric diagram



FROB (Forbus, 1980)



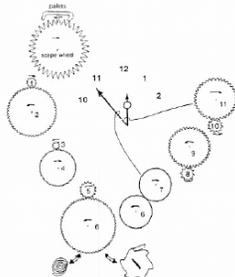
REGION 0  
left: SEGMENT 2  
right: SEGMENT 10  
up: SEGMENT 7  
down: SEGMENT 1  
class: REGION

SEGMENT 1  
up: REGION 0  
connecting-region: REGION 0  
class: SURFACE

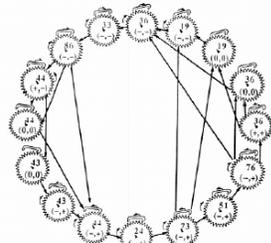
SEGMENT 2  
right: REGION 0  
left: SPATIUM-INDIGNITO  
connecting-region: REGION 0  
class: BORDER

SEGMENT 10  
left: REGION 0  
right: REGION 0  
class: FREE

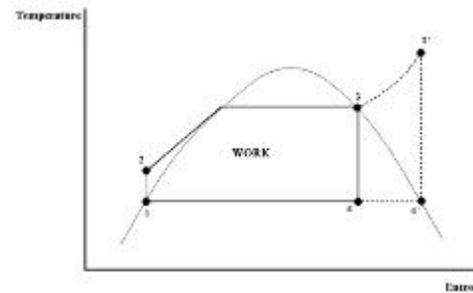
CLOCK Project: Example



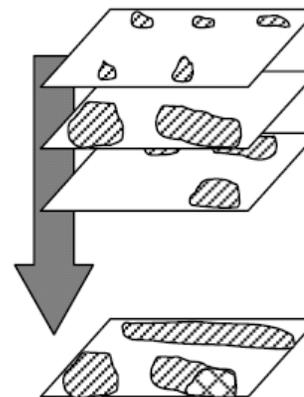
Input: Annotated diagram



Output: Set of possible behaviors, both desired and potential malfunctions



SKETCHY (Pisan, 1994)



Hydrography

Vegetation

Slope

Etc...

Combined  
Obstacle  
Overlay

GIS-based  
Trafficability  
Reasoner  
(Donlon &  
Forbus, 1999)

# Qualitative/Quantitative Representations in Psychology

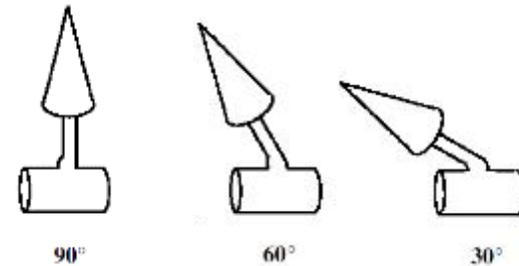
## Qualitative vs. Quantitative

Remembering locations

(Huttenlocher, Hedges, & Duncan, 1991;  
Holden et al. 2010)

Categorical perception of angle

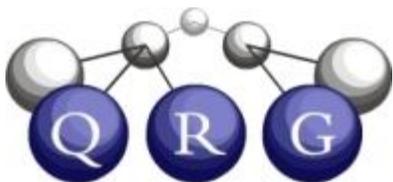
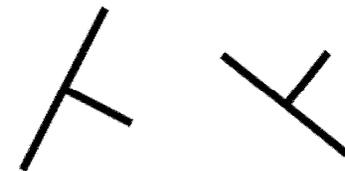
(Rosielle & Cooper, 2001)



## Qualitative advantages for complex reasoning

Smaller memory footprint

More robust across transformations



# Qualitative Spatial Relations

## Topological

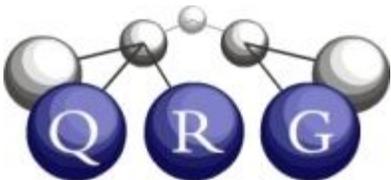
- Describes intersections and containment
- Computed automatically

## Positional

- Describes relative position
- Computed on demand

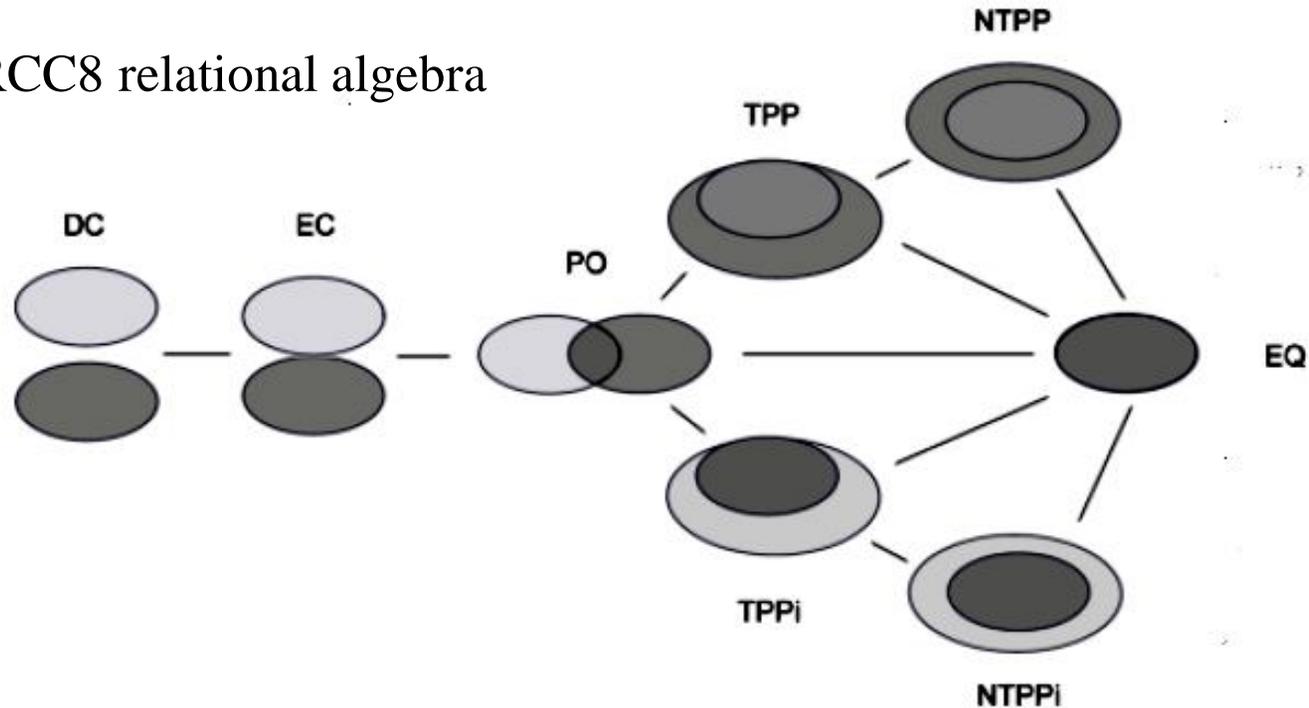
## Decompositional

- Describes cycles and edges within each glyph
- Computed on demand



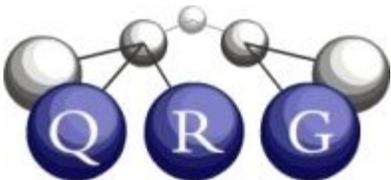
# Topological Relations

Cohn et al's RCC8 relational algebra



Provides natural vocabulary for some visual concepts

- Containment: NTPP, TPP
- Touching: PO, EC



# Using RCC8

Compute relationships directly from ink

Transitivity algebra unnecessary.

Need to be clever about noise.

Computed between every pair of glyphs on a layer

Incrementally updated when a glyph is moved or resized.

Only computed across layers on demand.

Internal uses

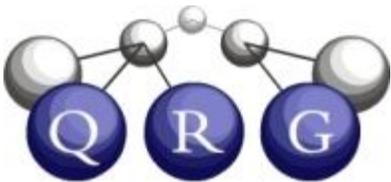
Controlling computation of other relations.

Positional relations aren't computed when there's containment.

Direct inference of other topological relations.

Convex hull topologicals can be queried for

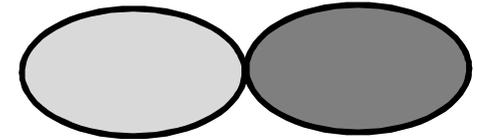
```
(hasRCC8Relation (ConvexHullFn Object-1)
  (ConvexHullFn Object-2)
  ?rel)
```



# Higher-Level Topological Relations

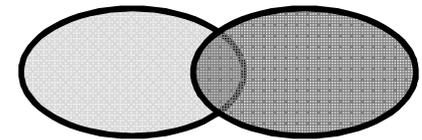
objectsIntersect

The ink of the two glyphs intersects



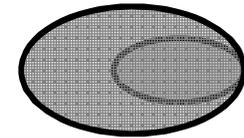
objectsOverlap

The interiors of the two glyphs overlap

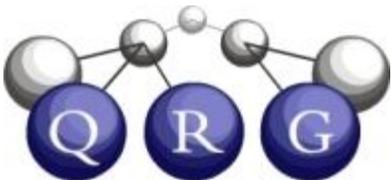


objectContains

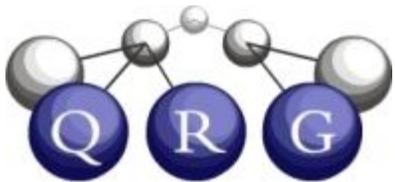
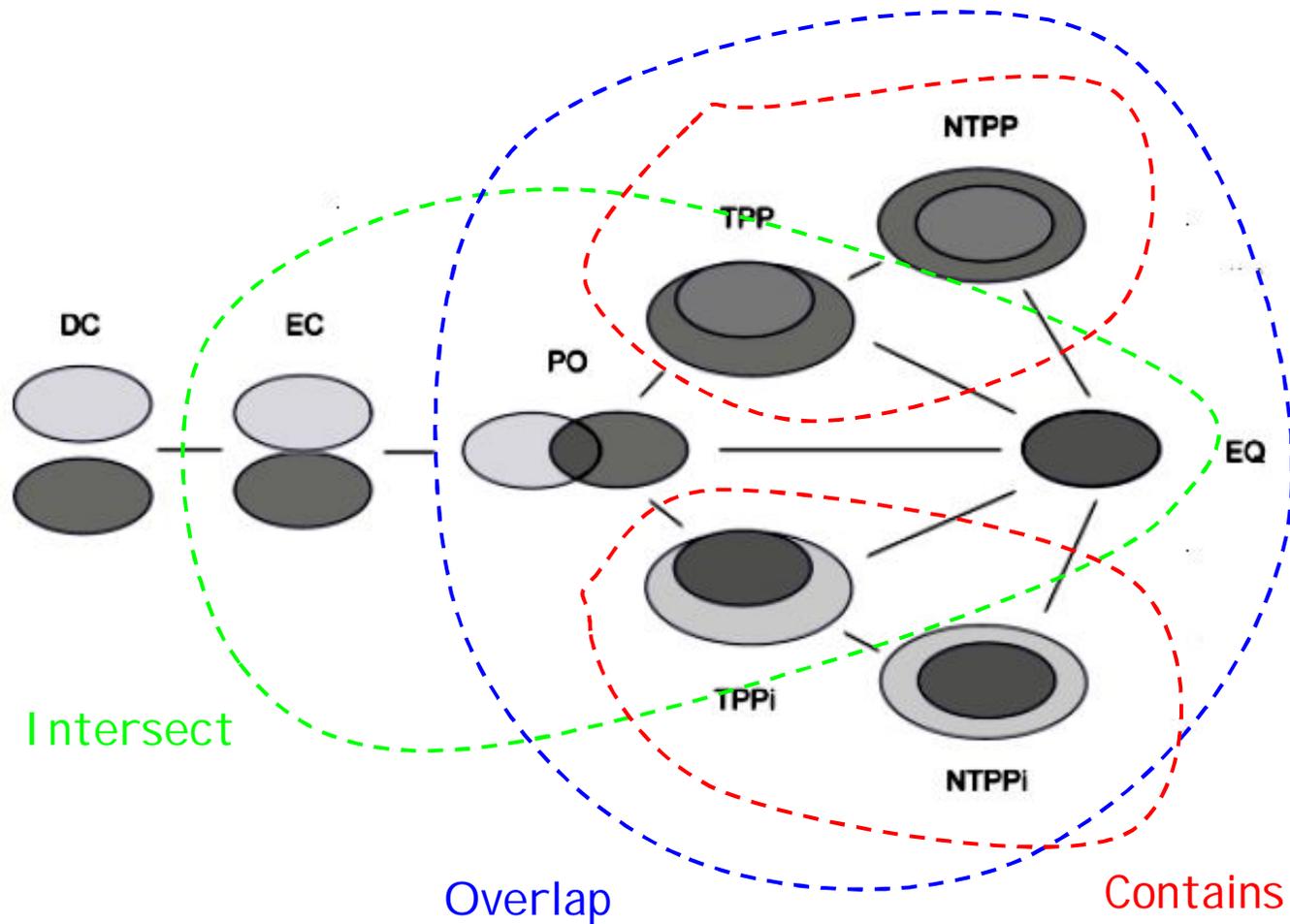
One glyph lies within another glyph's area



- Not mutually exclusive
- Used in comparison



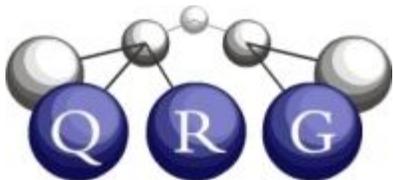
# RCC8 Conceptual Neighborhood



# Contained Glyph Groups

- When more than one glyph is NTTPi, TPPi of some other glyph

```
(ContainedGlyphGroupFn  
  (GlyphFn Object-9 User-Drawn-Sketch-Layer-1)  
  (TheList (GlyphFn Object-15 User-Drawn-Sketch-Layer-1)  
    (GlyphFn Object-16 User-Drawn-Sketch-Layer-1)  
    (GlyphFn Object-19 User-Drawn-Sketch-Layer-1)  
    (GlyphFn Object-20 User-Drawn-Sketch-Layer-1)))
```



# Connected Glyph Groups

- Set of glyphs connected via EC or PO

(ConnectedGlyphGroupFn

(TheList (GlyphFn Object-10 User-Drawn-Sketch-Layer-1)

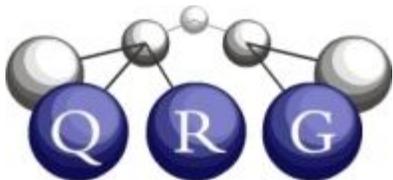
(GlyphFn Object-11 User-Drawn-Sketch-Layer-1)

(GlyphFn Object-12 User-Drawn-Sketch-Layer-1)

(GlyphFn Object-21 User-Drawn-Sketch-Layer-1)

(GlyphFn Object-22 User-Drawn-Sketch-Layer-1)

(GlyphFn Object-9 User-Drawn-Sketch-Layer-1)))



# Positional Relations

Provide qualitative position, orientation information with respect to global frame of reference

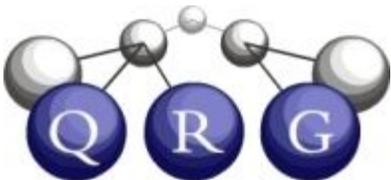
For glyphs, `rightOf`, `above`, `leftOf`, `below`

For contents, depends on genre and viewpoint

Physical/side: Same as glyphs

Geospatial/TopDown: `northOf`, `southOf`, `eastOf`, `westOf`

Abstract or Discrete: No implications for contents



# Local Relational Neighborhood Hypothesis

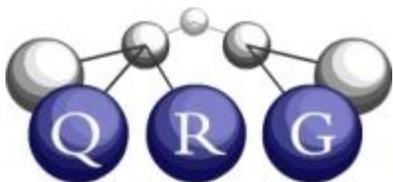
When to compute positional relations? Between every pair of glyphs on a layer, like RCC8?

- Bad idea! Loses locality

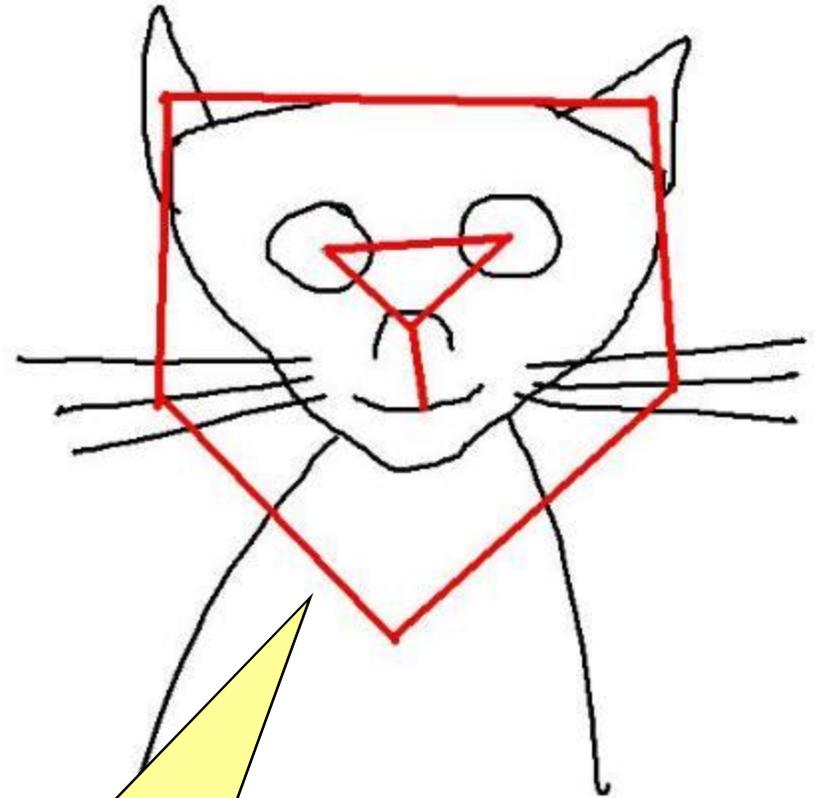
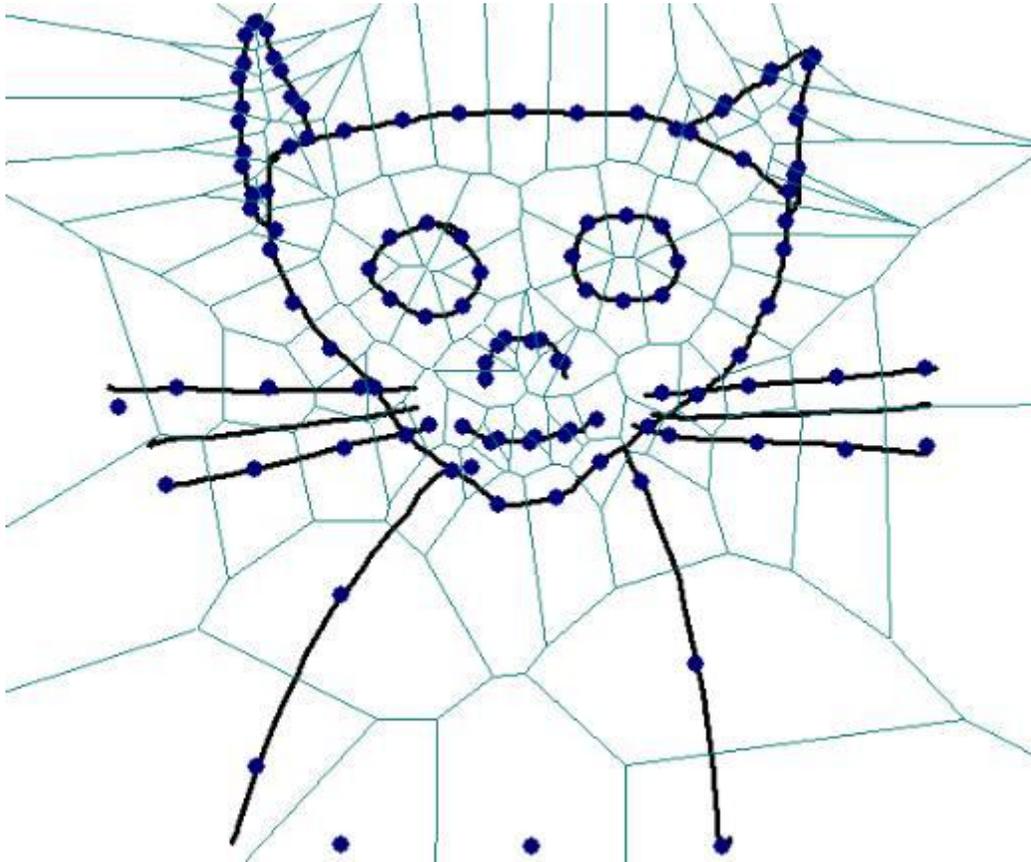
Idea: Network of positional relations should provide “framing effect” in visual structure.

Necessary condition: Glyphs must be *adjacent* in the sketch.

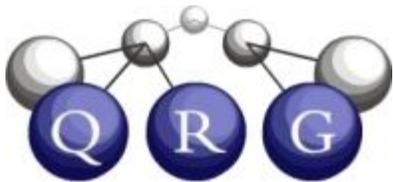
Hypothesis: This local neighborhood structure corresponds to default encoding method in human sketch perception.



# Voronoi adjacency guides positional relation finding



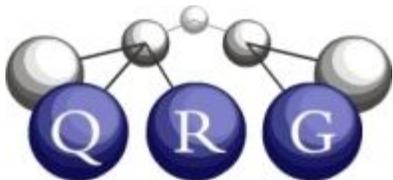
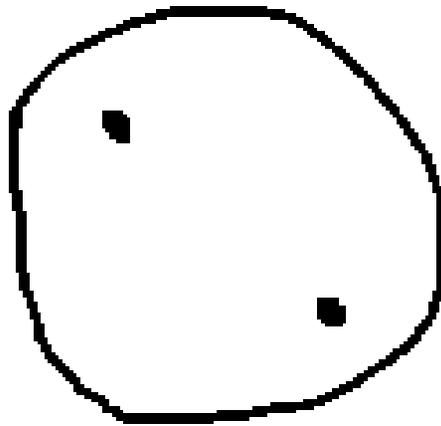
Positional relations only created between site-adjacent glyphs



# Frame of Reference (FoR) Relations

Glyphs create a *frame of reference* by which other glyphs' positions can be judged.

centeredOn, onRightHalfOf,  
onBottomHalfOf, etc.



# Decompositions

Hypothesis: *Shape* can be described in the same way as *space*.

Space: Qualitative spatial relations between glyphs

Shape: Qualitative spatial relations between parts of a glyph

Examples from TU  
Berlin corpus  
(Eitz et al. 2012;  
20,000 sketches  
spanning 250 concepts)

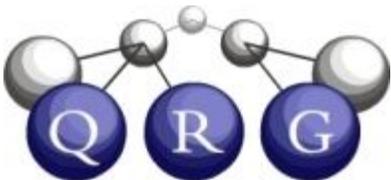
Two levels of decomposition:

Edges

Ink segmented based on sharp changes in curvature, junctions where multiple edges meet

Edge Cycles

Adjacent edges grouped to form cycles



# Decompositions

Hypothesis: *Shape* can be described in the same way as *space*.

Space: Qualitative spatial relations between glyphs

Shape: Qualitative spatial relations between parts of a glyph

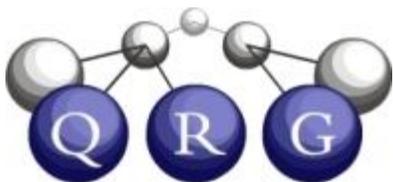
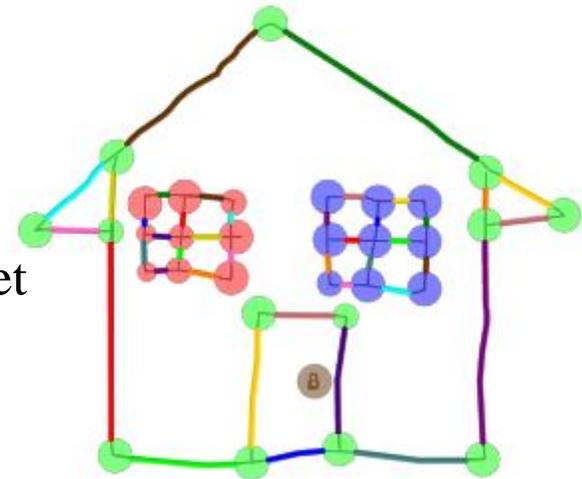
Two levels of decomposition:

## Edges

Ink segmented based on sharp changes in curvature, junctions where multiple edges meet

## Edge Cycles

Adjacent edges grouped to form cycles



# Decompositions

Hypothesis: *Shape* can be described in the same way as *space*.

Space: Qualitative spatial relations between glyphs

Shape: Qualitative spatial relations between parts of a glyph

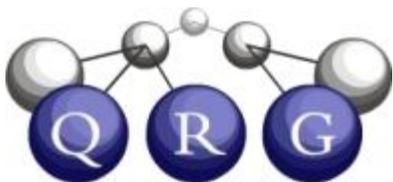
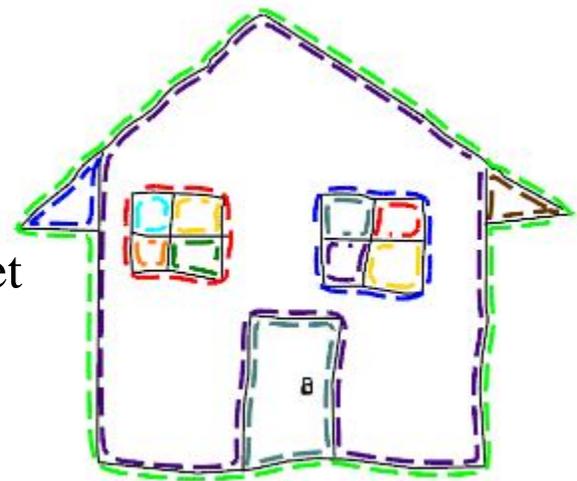
Two levels of decomposition:

Edges

Ink segmented based on sharp changes in curvature, junctions where multiple edges meet

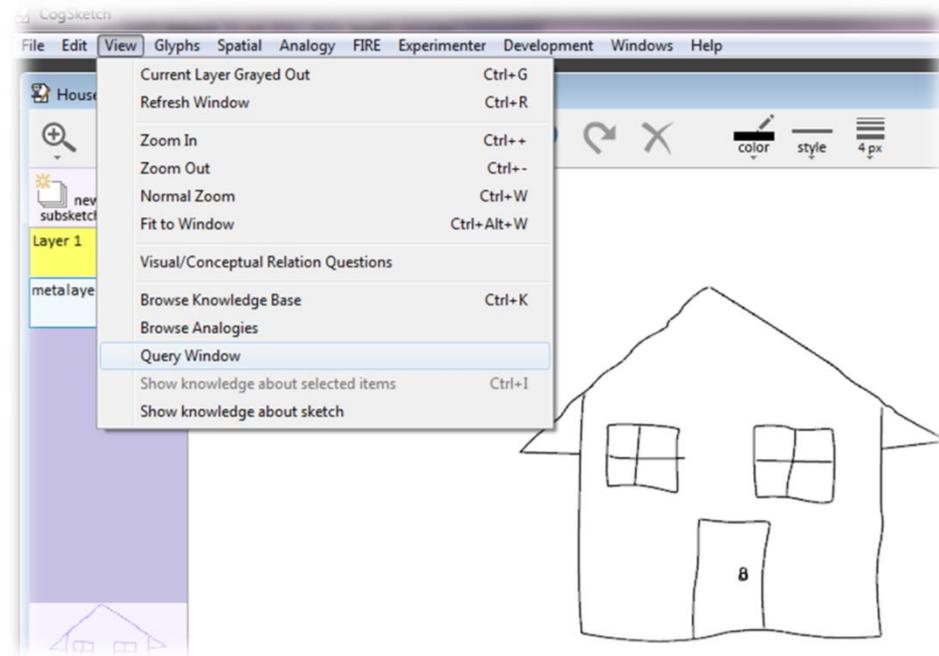
Edge Cycles

Adjacent edges grouped to form cycles



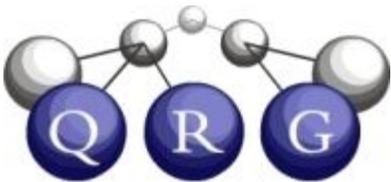
# Querying for Representations

...via Query Window



...via KQML interface

```
(query :sketch-id <sketch-id>
      :query <query>
      :num-responses :all)
```



# Query Window

**house**  
Case-3662670010

---

**Subsketch Subsketch 1**  
BCase-3662670010

**Layer Positional**  
ObjectL-9501

**Layer Voronoi**  
ObjectL-9500

**Layer Layer 1**  
ObjectL-8298

**House**  
Object-7

---

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM

Browse KB

Query / WM Fact Edit nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

(edgeCycleRepresentationsFor Object-7 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)

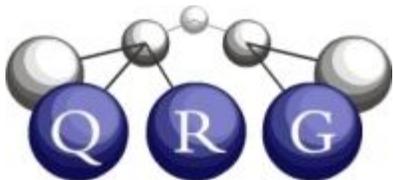
Context:

Facts:

Allow microtheory inheritance? (env)

Allow gens inferencing? (transitive)

Allow other kinds of inference? (infer)



# Query Window

house  
Case-3662670010

Subsketch Subsketch 1  
BCase-3662670010

Layer Positional  
ObjectL-9501

Layer Voronoi  
ObjectL-9500

Layer Layer 1  
ObjectL-8298

House  
Object-7

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM

Browse KB

Query / WM Fact Edit

nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

(edgeCycleRepresentationsFor Object-7 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)

Context: BCase-3662670010

Facts: all

Allow microtheory inheritance? (env)

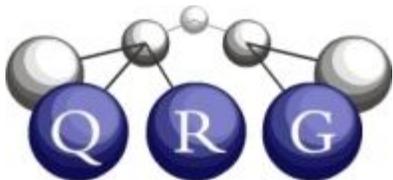
Allow gens inferencing? (transitive)

Allow other kinds of inference? (infer)

Query using fire:ask Query using fire:query Tell Untell

Object name or glyph name

- If unbound, will query all glyphs in subsketch



# Query Window

house  
Case-3662670010

Subsketch Subsketch 1  
BCase-3662670010

Layer Positional  
ObjectL-9501

Layer Voronoi  
ObjectL-9500

Layer Layer 1  
ObjectL-8298

House  
Object-7

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM

Browse KB

Query / WM Fact Edit

nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

(edgeCycleRepresentationsFor Object-7 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)

Angle threshold modifier

- Adjusts flexibility in identifying straight or right angles
- Range: [-1, 2]

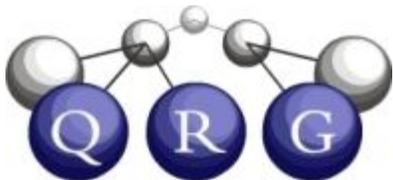
Facts: all

Allow microtheory inheritance? (env)

Allow gens inferencing? (transitive)

Allow other kinds of inference? (infer)

Query using fire:ask Query using fire:query Tell Untell



# Query Window

house  
Case-3662670010

Subsketch Subsketch 1  
BCase-3662670010

Layer Positional  
ObjectL-9501

Layer Voronoi  
ObjectL-9500

Layer Layer 1  
ObjectL-8298

House  
Object-7

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM

Browse KB

Query / WM Fact Edit

nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

(edgeCycleRepresentationsFor Object-7 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)

Size threshold modifier

- Adjusts flexibility for calling things “same size”
- Range: [-1, 2]

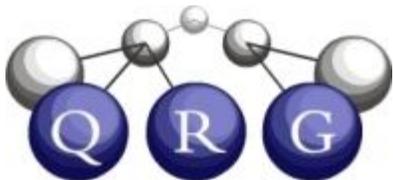
Facts: all

Allow microtheory inheritance? (env)

Allow gens inferencing? (transitive)

Allow other kinds of inference? (infer)

Query using fire:ask Query using fire:query Tell Untell



# Query Window

**house**  
Case-3662670010

---

**Subsketch Subsketch 1**  
BCase-3662670010

**Layer Positional**  
ObjectL-9501

**Layer Voronoi**  
ObjectL-9500

**Layer Layer 1**  
ObjectL-8298

**House**  
Object-7

---

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM

Browse KB

Query / WM Fact Edit nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

(edgeCycleRepresentationsFor Object-7 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)

Context: BCase-3662670010

Facts:

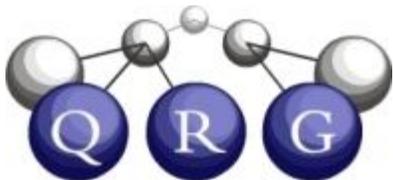
Allow microtheory inheritance? (env)

Allow gens inferencing? (transitive)

Allow other kinds of inference? (infer)

Query using fire:ask    Query using fire:query    Tell    Untell

Gets bound to a microtheory containing representations about edge-cycles in the glyph



# Query Window

**house**  
Case-3662670010

---

**Subsketch Subsketch 1**  
BCase-3662670010

**Layer Positional**  
ObjectL-9501

**Layer Voronoi**  
ObjectL-9500

**Layer Layer 1**  
ObjectL-8298

**House**  
Object-7

---

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM

Browse KB

Query / WM Fact Edit nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

(edgeCycleRepresentationsFor Object-7 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)

Gets bound to the set of edge-cycle names

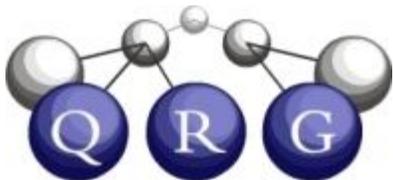
Context:

Facts:

Allow microtheory inheritance? (env)

Allow gens inferencing? (transitive)

Allow other kinds of inference? (infer)



# Query Window

**house**  
Case-3662670010

---

**Subsketch Subsketch 1**  
BCase-3662670010

**Layer Positional**  
ObjectL-9501

**Layer Voronoi**  
ObjectL-9500

**Layer Layer 1**  
ObjectL-8298

**House**  
Object-7

---

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM

Browse KB

Query / WM Fact Edit nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

(edgeCycleRepresentationsFor Object-7 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)

**Context:** BCase-3662670010

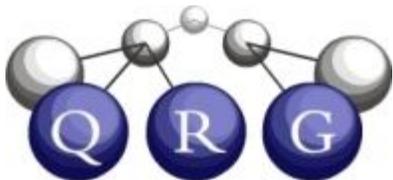
**Facts:**

Allow microtheory inheritance? (env)

Allow gens inferencing? (transitive)

Allow other kinds of inference? (infer)

Gets bound to the number of facts generated



# Query Window

**house**  
Case-3662670010

**Subsketch Subsketch 1**  
BCase-3662670010

**Layer Positional**  
ObjectL-9501

**Layer Voronoi**  
ObjectL-9500

**Layer Layer 1**  
ObjectL-8298

**House**  
Object-7

**Clarify Glyph Relationships**

- Query / WM Fact Edit
- KB Fact Edit
- Analogy
- Refresh Object List
- Browse all WM
- Browse KB

**Query / WM Fact Edit** nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

```
(edgeCycleRepresentationsFor Object-7 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)
```

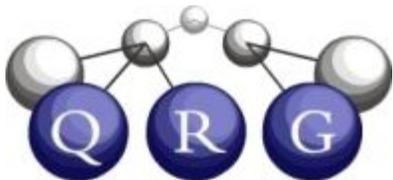
Context: BCase-3662670010

Facts:

- Allow microtheory inheritance? (env)
- Allow genls inferencing? (transitive)
- Allow other kinds of inference? (infer)

Query using fire:ask   Query using fire:query   Tell   Untell

Query in the Subsketch context



# Query Window

**house**  
Case-3662670010

---

**Subsketch Subsketch 1**  
BCase-3662670010

**Layer Positional**  
ObjectL-9503

**Layer Voronoi**  
ObjectL-9502

**Layer Layer 1**  
ObjectL-8298

**House**  
Object-7

---

**Clarify Glyph Relationships**

- Query / WM Fact Edit
- KB Fact Edit
- Analogy
- Refresh Object List
- Browse all WM 
- Browse KB 

**Query / WM Fact Edit** nuSketch Reasoner #27  
opencyc4 KB (HORCRUXWHOA)

```
(edgeCycleRepresentationsFor Object-7 0 0
?edge-cycle-microtheory ?edge-cycles ?n-facts)
action = ask
context = BCase-3662670010; facts = all, env, infer
```

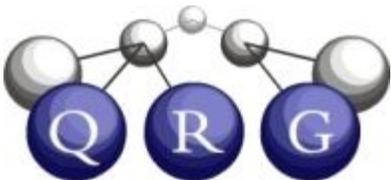
**Answers:**

**in BCase-3662670010:**

```
? A X (edgeCycleRepresentationsFor Object-7 0 0
(EdgeCycleFactsMtFn Object-7)
(TheSet EdgeCycle-61107 EdgeCycle-61111 EdgeCycle-61105
EdgeCycle-61103 (PerimeterEdgeCycleFn ECO-2851957)
EdgeCycle-61099 EdgeCycle-61097 EdgeCycle-61098
EdgeCycle-61100 (PerimeterEdgeCycleFn ECO-2851959)
EdgeCycle-61110 EdgeCycle-61104 EdgeCycle-61108
EdgeCycle-61106 (PerimeterEdgeCycleFn ECO-2851960)) 271)
```

Refresh the object pane

Return to Query/Edit Page



# Query Window

**Layer Positional**  
ObjectL-9501

**Layer Voronoi**  
ObjectL-9500

**Layer Layer 1**  
ObjectL-8298

**House**  
Object-7  
 [\(PerimeterEdgeCycleFn ECO-2851957\)](#)  
 [EdgeCycle-61106](#)  
 [EdgeCycle-61108](#)  
 [...](#)

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM 

Browse KB 

**Query / WM Fact Edit** nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

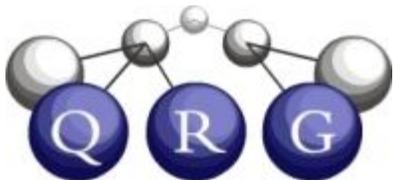
```
(edgeCycleRepresentationsFor Object-7 0 0  
?edge-cycle-microtheory ?edge-cycles ?n-facts)  
action = ask  
context = BCase-3662670010; facts = all, env, infer
```

**Answers:**

```
ntationsFor Object-7 0 0  
MtFn Object-7)  
le-61107 EdgeCycle-61111 EdgeCycle-61105  
03 (PerimeterEdgeCycleFn ECO-2851957)  
99 EdgeCycle-61097 EdgeCycle-61098  
EdgeCycle-61100 (PerimeterEdgeCycleFn ECO-2851959)  
EdgeCycle-61110 EdgeCycle-61104 EdgeCycle-61108  
EdgeCycle-61106 (PerimeterEdgeCycleFn ECO-2851960)) 271)
```

[Return to Query/Edit Page](#)

Click to view facts about a perceptual entity



# Query Window

**Layer Positional**  
ObjectL-9501

**Layer Voronoi**  
ObjectL-9500

**Layer Layer 1**  
ObjectL-8298

**House**  
Object-7

- [PerimeterEdgeCycleFn ECO-2851957]**
- EdgeCycle-61106**
- EdgeCycle-61108**
- EdgeCycle-61104**
- EdgeCycle-61110**
- EdgeCycle-61100**
- EdgeCycle-61098**

Clarify Glyph Relationships

Query / WM Fact Edit

KB Fact Edit

Analogy

Refresh Object List

Browse all WM 

Browse KB 

Query / WM Fact Edit

nuSketch Reasoner #26  
opencyc4 KB (HORCRUXWHOA)

Enter a fact or query here:

(tracingEdgeRepresentationsForEdgeCycle (PerimeterEdgeCycleFn ECO-2851957) 0 0 ?edge-cycle-microtheory ?edge-cycles ?n-facts)

When querying for edge representations, pass in an edge cycle name

Context: (EdgeCycleFactsMtFn Object-7)

Facts: all

Query in the edge-cycle microtheory

- Allow microtheory inheritance? (env)
- Allow gens inferencing? (transitive)
- Allow other kinds of inference? (infer)

Query using fire:ask

Query using fire:query

Tell

Untell



# Edge-Level Representations

## Edge Attributes

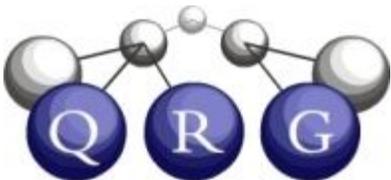
Straight, Curved, Ellipse

Curve details: Concave/Convex,  
Major/Minor/Semicircle arcs

Orientation

Relative length

```
(isa Edge-104451 StraightEdge)  
(isa Edge-104451 VerticalEdge)  
(lengthMedium Edge-104451)
```



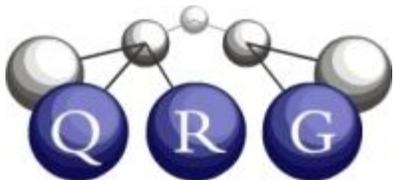
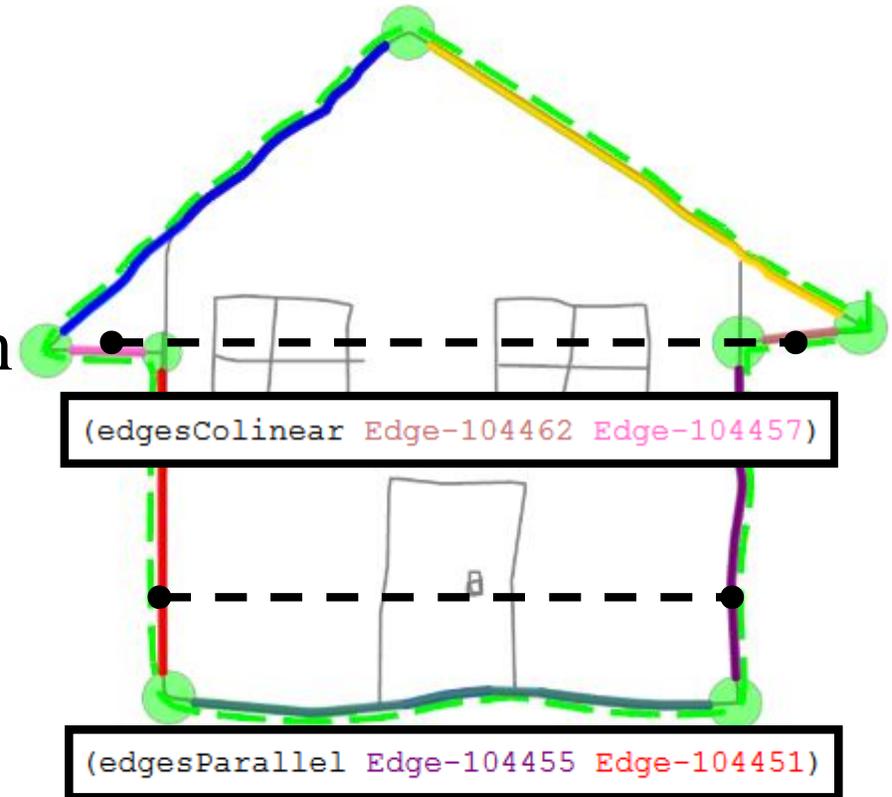
# Edge-Level Representations

## Edge Attributes

Straight, Curved, Ellipse  
Curve details: Concave/Convex,  
Major/Minor/Semicircle arcs  
Orientation  
Relative length

## Relative Orientation/Position

Parallel, Perpendicular,  
Collinear



# Edge-Level Representations

## Edge Attributes

Straight, Curved, Ellipse

Curve details: Concave

Major/Minor/Semi

Orientation

Relative length

## Relative Orientation/Position

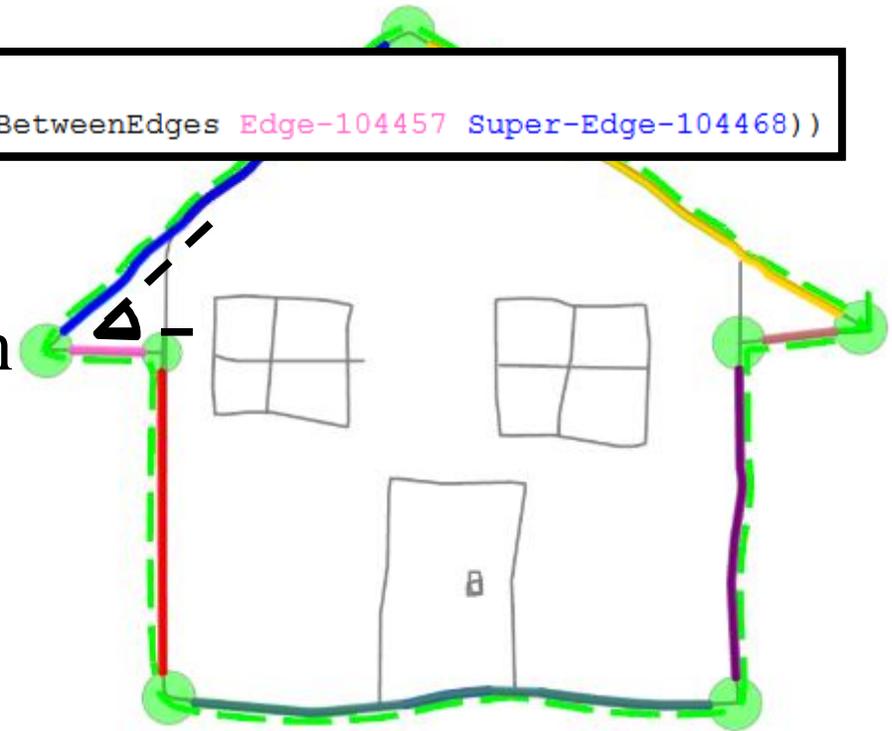
Parallel, Perpendicular,  
Collinear

## Corner Attributes

Convex, Concave

Right, Acute, Obtuse

```
(acuteCorner  
(convexAngleBetweenEdges Edge-104457 Super-Edge-104468))
```



# Edge-Level Representations

## Edge Attributes

Straight, Curved,  
Curve details: Co  
Major/Minor/Semicircle arcs

```
(cycleAdjacentAngles  
(concaveAngleBetweenEdges Edge-104451 Edge-104457)  
(convexAngleBetweenEdges Edge-104457 Super-Edge-104468))
```

Orientation  
Relative length

## Relative Orientation/Position

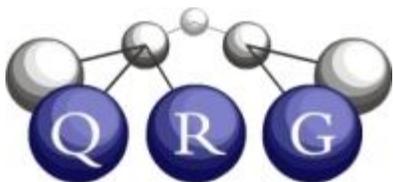
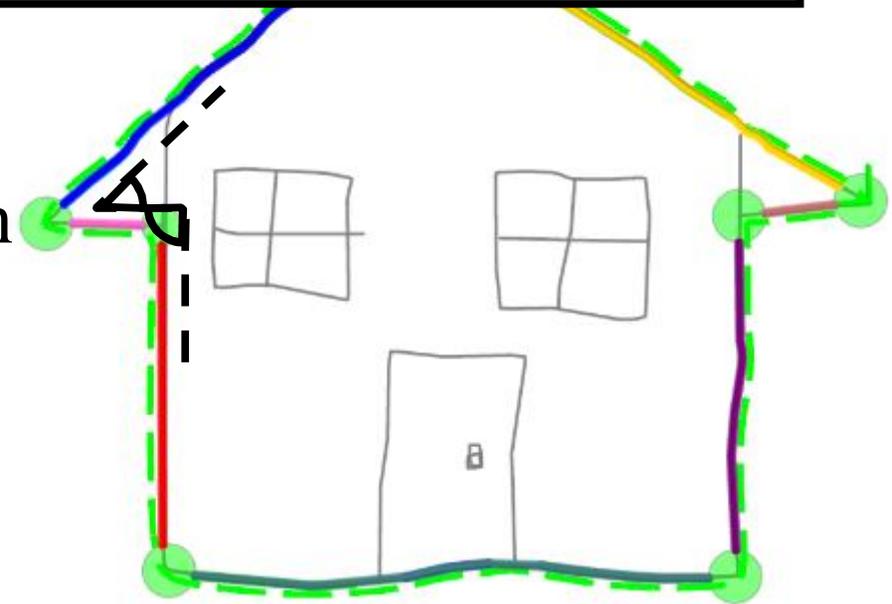
Parallel, Perpendicular,  
Collinear

## Corner Attributes

Convex, Concave  
Right, Acute, Obtuse

## Corner Relations

Corner adjacency



# Edge Cycle-Level Representations

## Shape Attributes

Edge-aggregated: Straight, Curved, Axis-aligned,

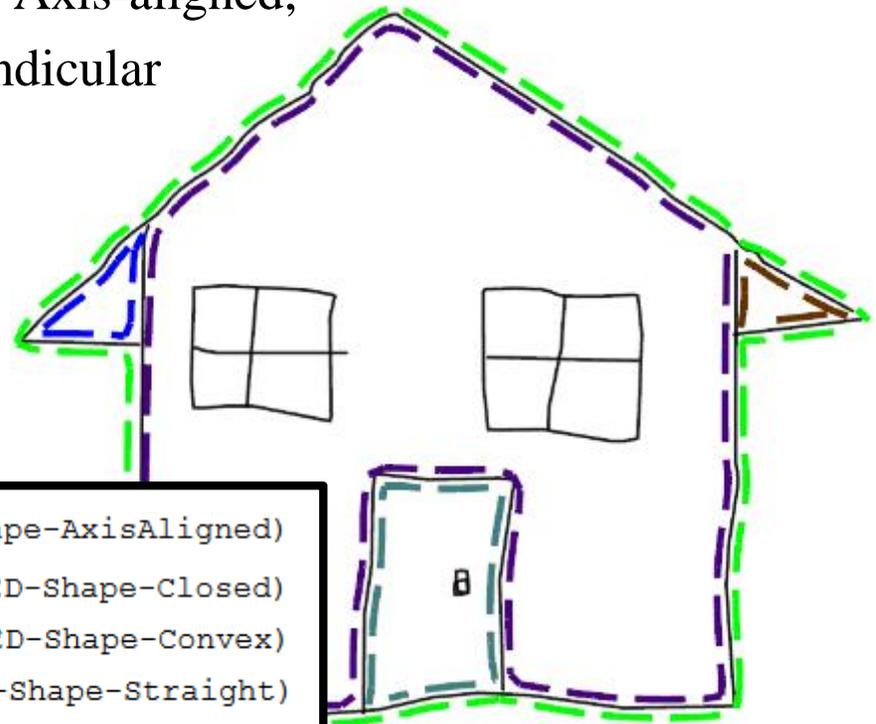
Corner-aggregated: Convex, Perpendicular

Closed, Partially-Closed, Open

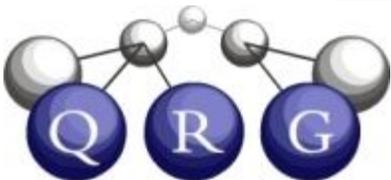
Major axis

Axis of symmetry

Relative area, edge-complexity



```
(isa EdgeCycle-61103 2D-Shape-AxisAligned)
  (isa EdgeCycle-61103 2D-Shape-Closed)
    (isa EdgeCycle-61103 2D-Shape-Convex)
      (isa EdgeCycle-61103 2D-Shape-Straight)
        (isa EdgeCycle-61103 2D-Shape-VerticalMajorAxis)
          (areaTiny EdgeCycle-61103)
            (edgeComplexityLow EdgeCycle-61103)
```



# Edge Cycle-Level Representations

## Shape Attributes

Edge-aggregated: Straight, Curved, Axis-aligned,

Corner-aggregated: Convex, Perpendicular

Closed, Partially-Closed, Open

Major axis

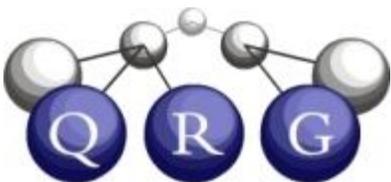
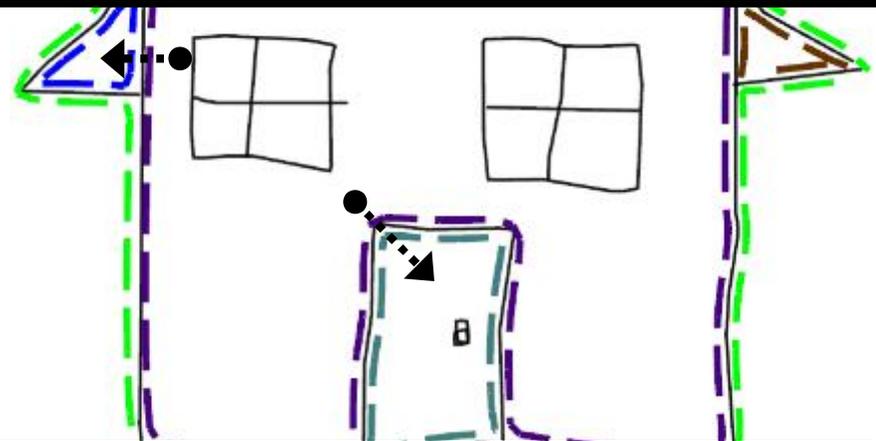
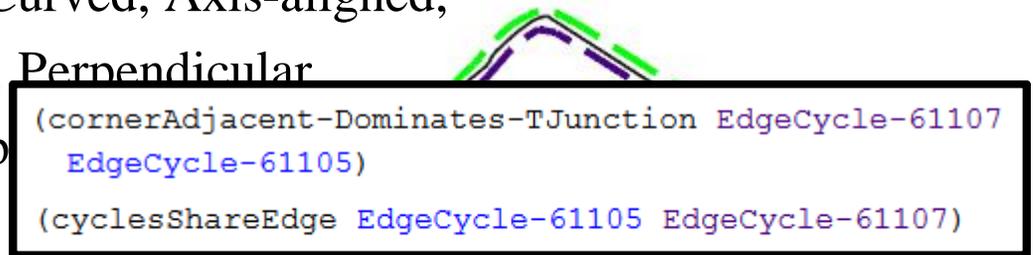
Axis of symmetry

Relative area, edge-complexity

## Connection Relations

Shared edge/junction

Adjacent corner shape (T, Y,  $\wedge$ , L)



# Edge Cycle-Level Representations

## Shape Attributes

Edge-aggregated: Straight, Curved, Axis-aligned,

Corner-aggregated: Convex, Perpendicular

Closed, Partially-Closed, Open

Major axis

Axis of symmetry

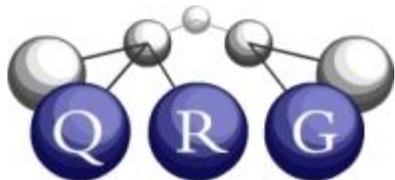
Relative area, edge-complexity

## Connection Relations

Shared edge/junction

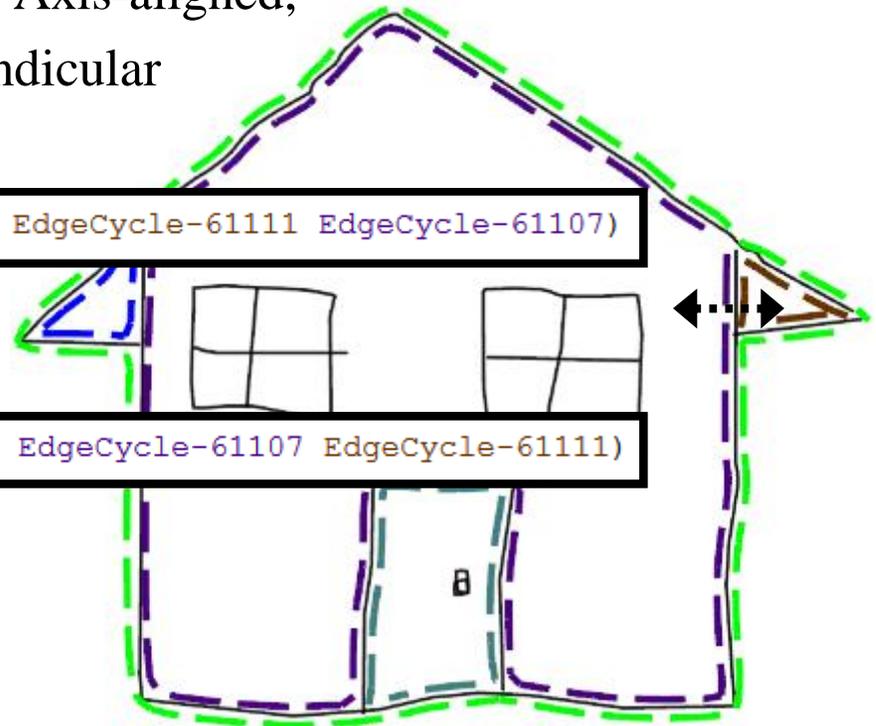
Adjacent corner shape (T, Y,  $\wedge$ , L)

## Positional Relations



`(rightOf EdgeCycle-61111 EdgeCycle-61107)`

`(enclosesVertically EdgeCycle-61107 EdgeCycle-61111)`



# Edge Cycle-Level Representations

## Shape Attributes

Edge-aggregated: Straight, Curved, Axis-aligned,

Corner-aggregated: Convex, Perpendicular

Closed, Partially-Closed, Open

Major axis

Axis of symmetry

Relative area, edge-complexity

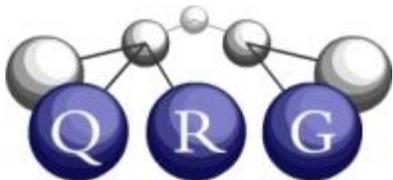
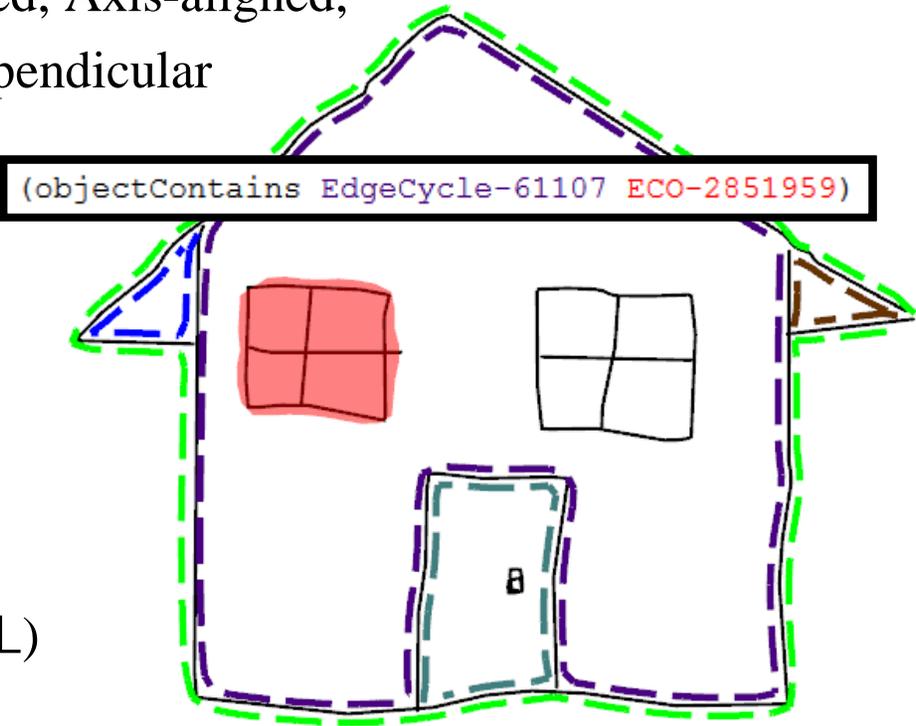
## Connection Relations

Shared edge/junction

Adjacent corner shape (T, Y,  $\wedge$ , L)

## Positional Relations

## Containment

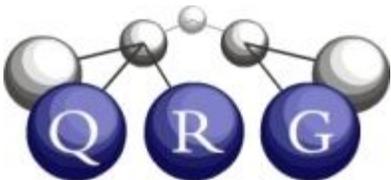
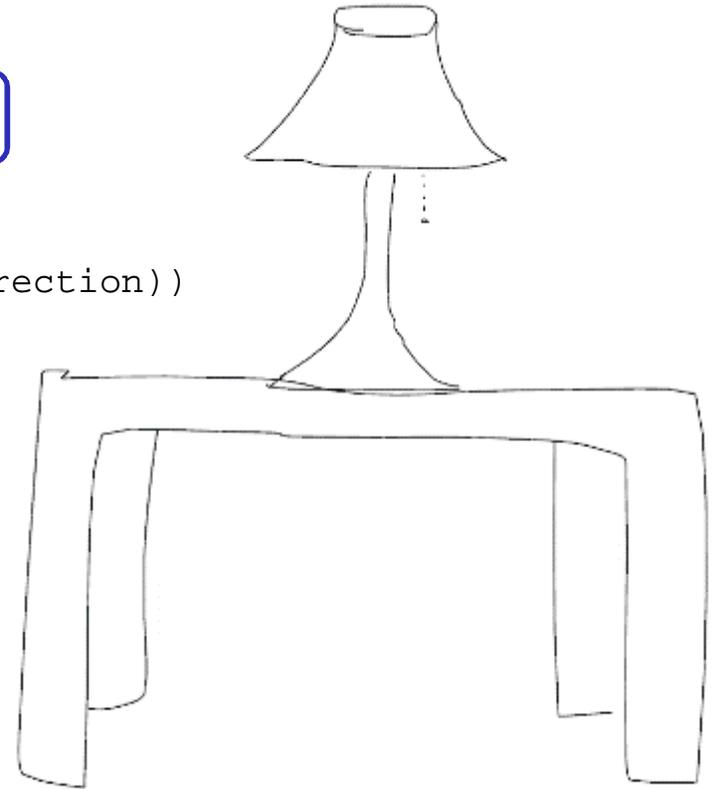


# Surface Contact Detection

Edges in decompositions are used to compute a more stable contact edge or point between two glyphs

Query in the Subsketch context

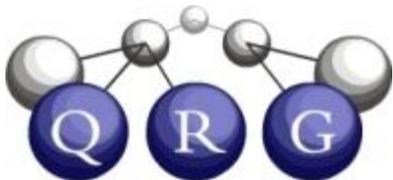
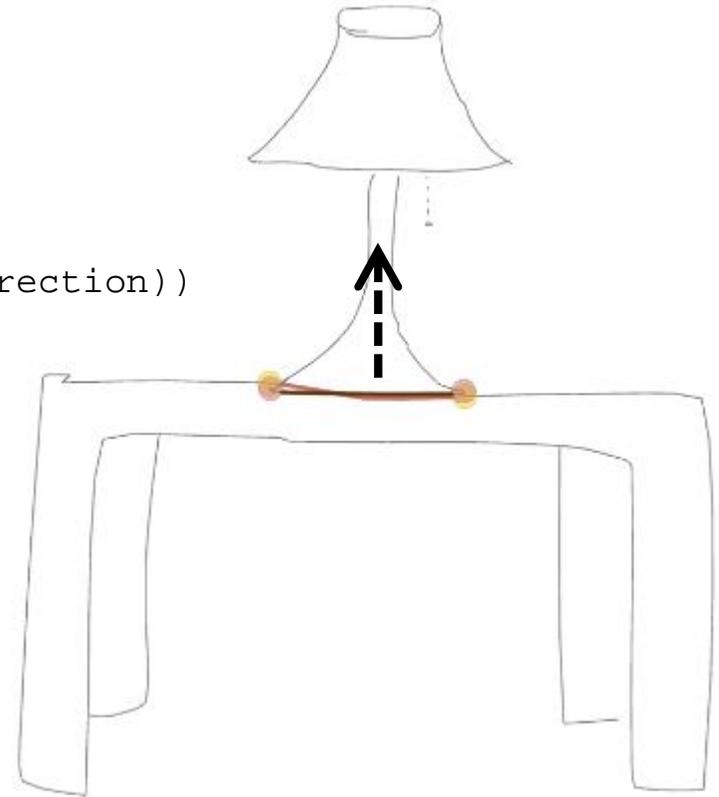
```
(ist-Information BCase-3662865065  
  (surfaceContactDirection Table TableLamp ?direction))
```



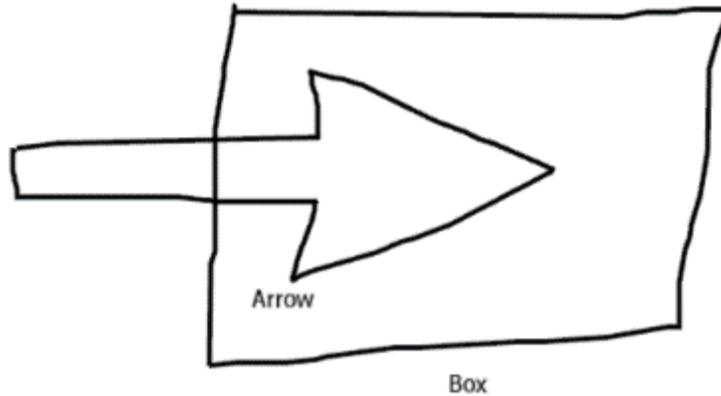
# Surface Contact Detection

Edges in decompositions are used to compute a more stable contact edge or point between two glyphs

```
(ist-Information BCase-3662865065  
  (surfaceContactDirection Table TableLamp ?direction))  
  
(surfaceContactDirection Table TableLamp Up)
```



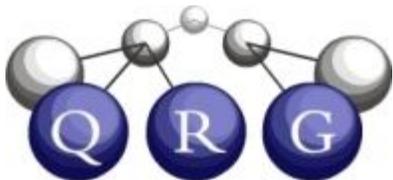
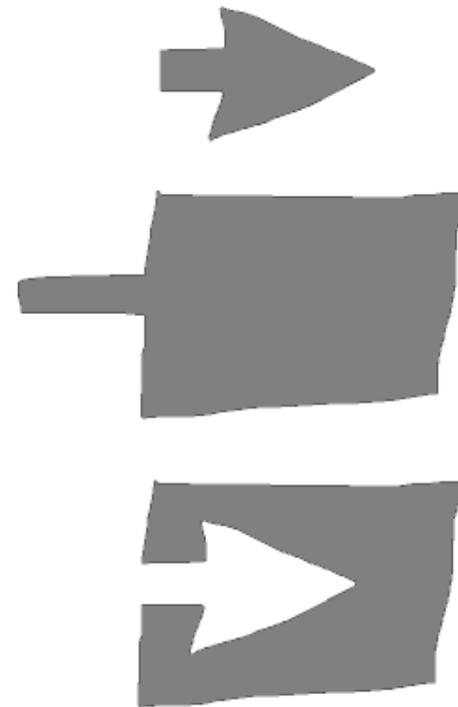
# Generating New Glyphs from Visual Relationships



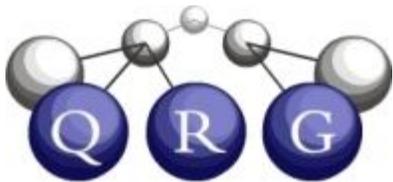
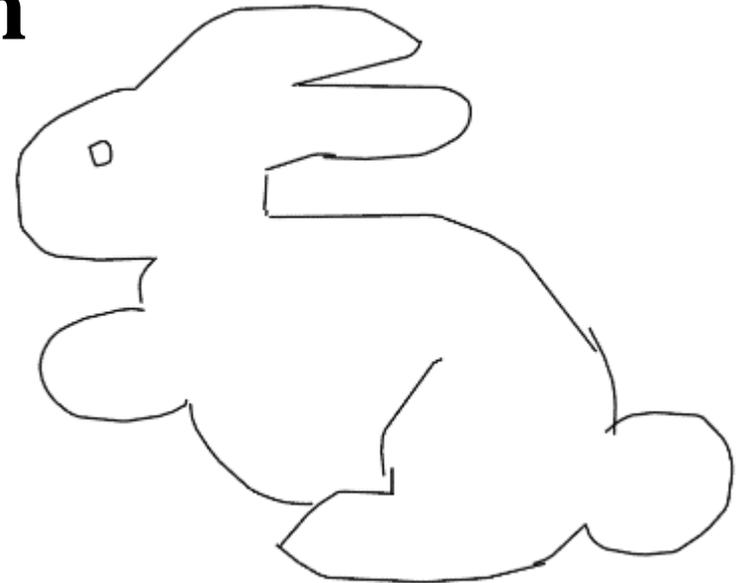
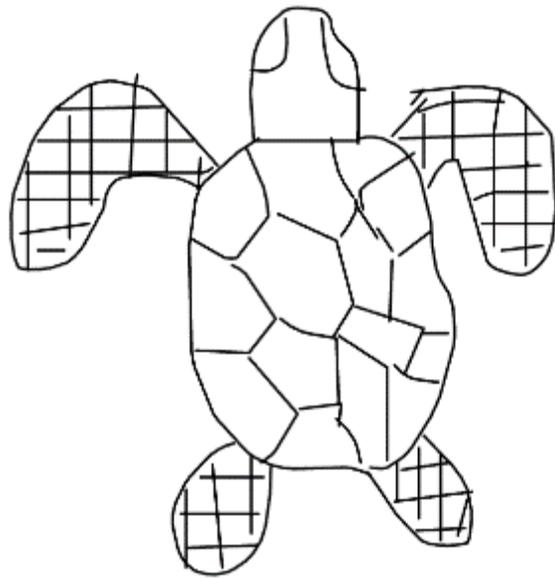
glyphFromGlyphIntersection

glyphFromGlyphUnion

glyphFromGlyphDifference

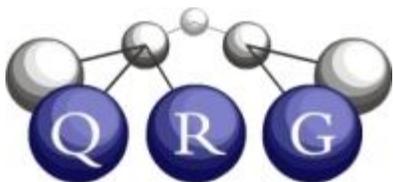
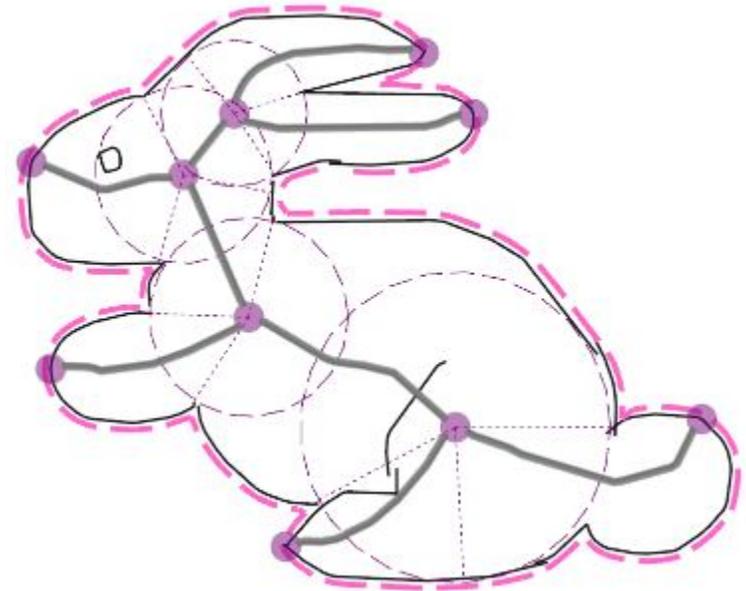


# Experimental CogSketch



# Medial Axis Transform (MAT)

1. The points on the interior of a closed shape that have more than one closest point on the exterior
2. The radius function  $R$ : For every point, the distance from the exterior



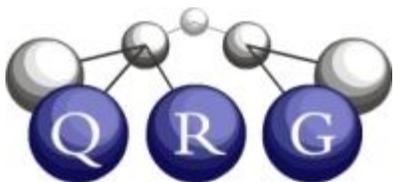
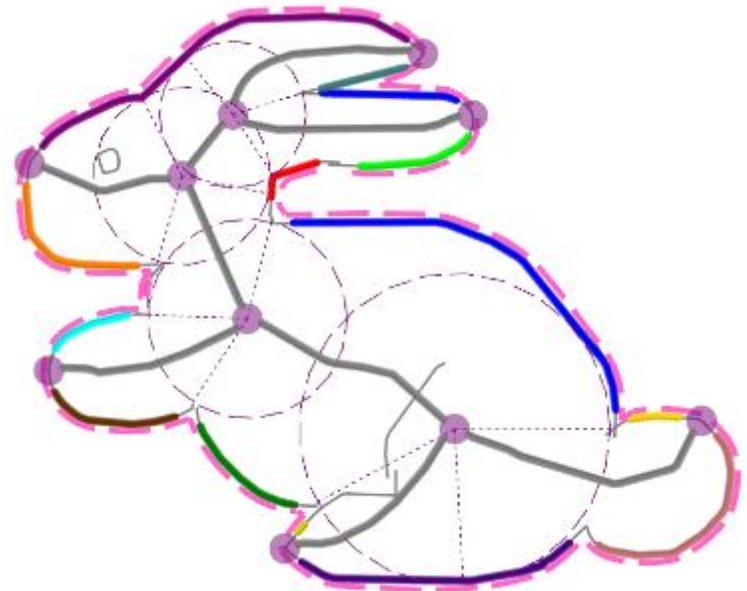
# Medial Axis Transform (MAT)

1. The points on the interior of a closed shape that have more than one closest point on the exterior
2. The radius function  $R$ : For every point, the distance from the exterior

Sketched ink produces “hairy” MATs.

Filter using exterior segmentation

(Bai et al., 2007)



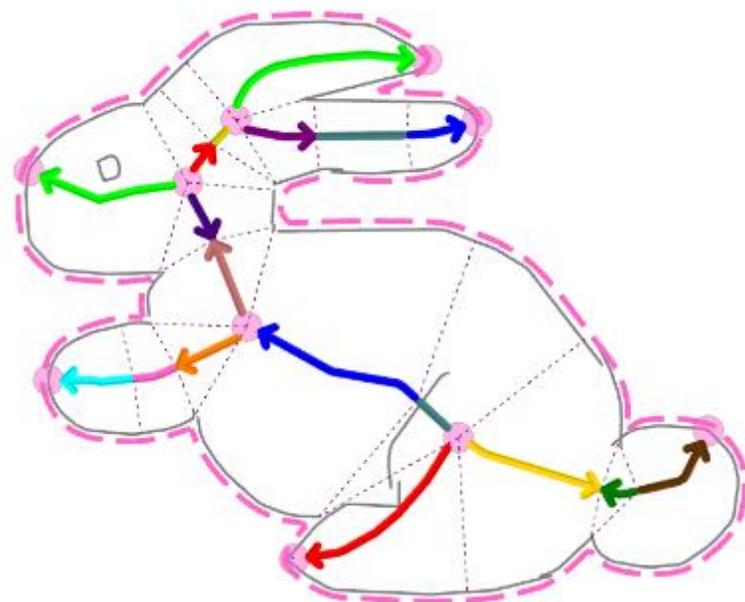
# Medial Axis Transform (MAT)

1. The points on the interior of a closed shape that have more than one closest point on the exterior
2. The radius function  $R$ : For every point, the distance from the exterior

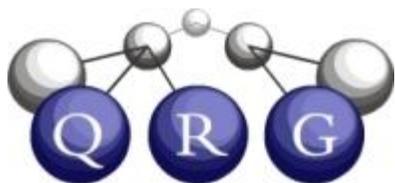
Sketched ink produces “hairy” MATs.

Filter using exterior segmentation

(Bai et al., 2007)



*Shock graphs* (Siddiqi et al. 1999) carve the MAT at qualitative changes in  $R$

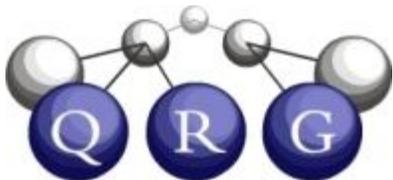
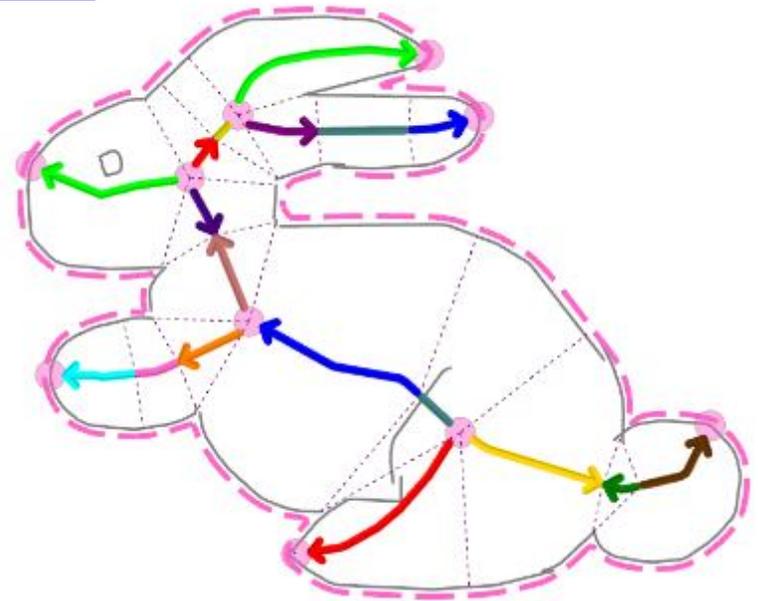


# Querying for MAT Representations

Query in the context containing edge-cycle facts

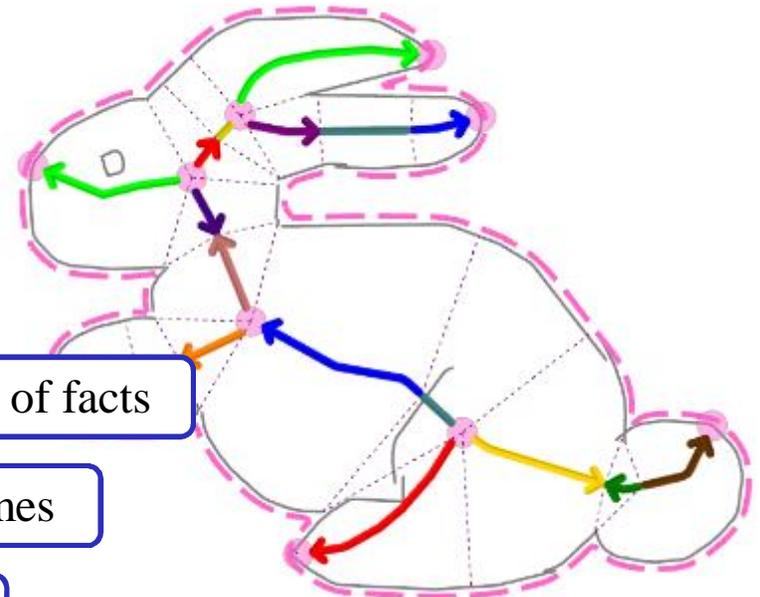
```
(list-Information (EdgeCycleFactsMtFn Rabbit)  
  (medialAxisRepresentationsForEdgeCycle  
    (PerimeterEdgeCycleFn ECO-3735956) ?angle-mod  
    ?size-mod ?mat ?nt ?edges ?n-facts)
```

Pass in an edge-cycle. Must be 2D-Shape-Closed or 2D-Shape-PartiallyClosed



# Querying for MAT Representations

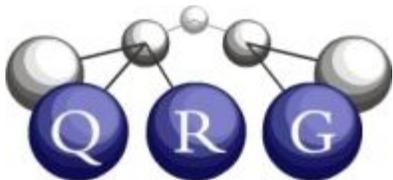
```
(ist-Information (EdgeCycleFactsMtFn Rabbit)  
  (medialAxisRepresentationsForEdgeCycle  
    (PerimeterEdgeCycleFn ECO-3735956) ?angle-mod  
    ?size-mod ?mat-mt ?edges ?n-facts)
```



Gets bound to the number of facts

Gets bound to the list of MAT edge names

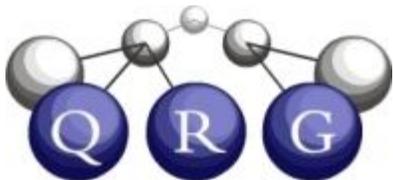
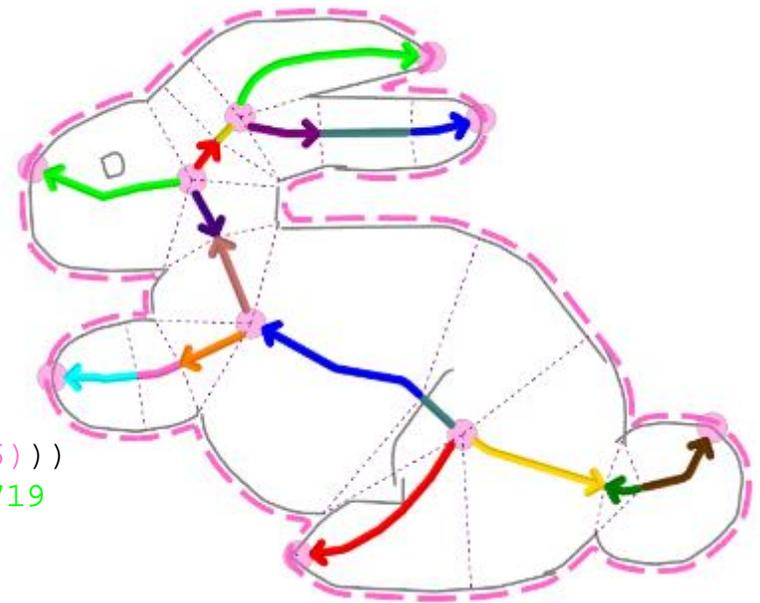
Gets bound to a microtheory containing the facts



# Querying for MAT Representations

```
(ist-Information (EdgeCycleFactsMtFn Rabbit)
  (medialAxisRepresentationsForEdgeCycle
    (PerimeterEdgeCycleFn ECO-3735956) ?angle-mod
    ?size-mod ?mat-mt ?edges ?n-facts)
```

```
(medialAxisRepresentationsForEdgeCycle
  (PerimeterEdgeCycleFn ECO-3735956) 0 0
  (DirectedEdgeFactsMtFn
    (MedialAxisFn (PerimeterEdgeCycleFn ECO-3735956)))
  (TheSet MedialAxisEdge-106721 MedialAxisEdge-106719
    MedialAxisEdge-106717 MedialAxisEdge-106715
    MedialAxisEdge-106713 MedialAxisEdge-106711
    MedialAxisEdge-106709 MedialAxisEdge-106707
    MedialAxisEdge-106705 MedialAxisEdge-106703
    MedialAxisEdge-106701 MedialAxisEdge-106699
    MedialAxisEdge-106697 MedialAxisEdge-106695
    MedialAxisEdge-106693 MedialAxisEdge-106691
    MedialAxisEdge-106689 MedialAxisEdge-106687) 267)
```



# MAT Representations

## Standard Edge Attributes

## Edge Radius Attributes

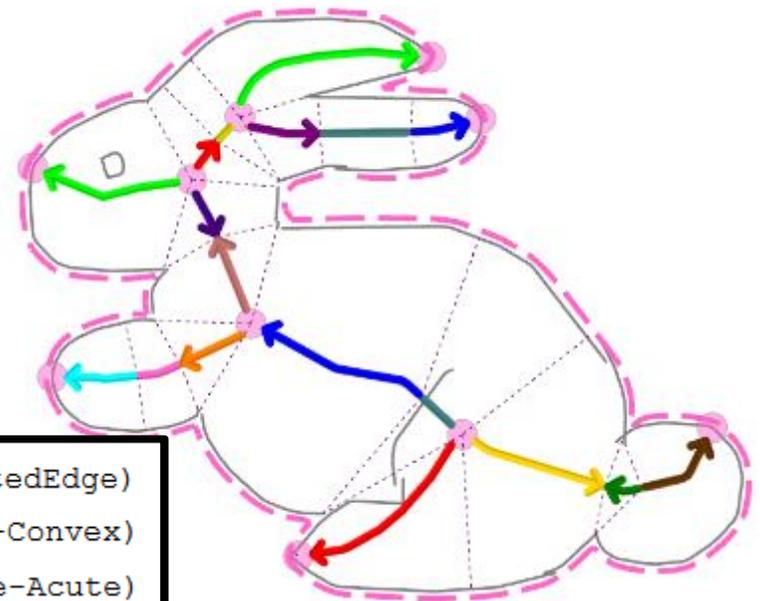
Directed/Undirected ( $R' \leq/\approx 0$ )

Concave/Linear/Convex ( $R'' >/\approx/< 0$ )

Obtuse/Right/Acute ( $R' >/\approx/< -1$ )

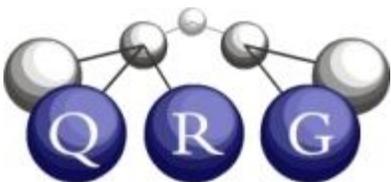
Source/Sink edge

Relative thickness



```
(isa MedialAxisEdge-106689 DirectedEdge)
(isa MedialAxisEdge-106689 MedialAxisEdge-Convex)
(isa MedialAxisEdge-106689 MedialAxisEdge-Acute)
```

```
(isa MedialAxisEdge-106687 UndirectedEdge)
(isa MedialAxisEdge-106687 SourceEdge)
```



# MAT Representations

## Standard Edge Attributes

### Edge Radius Attributes

Directed/Undirected

Concave/Linear/Convex

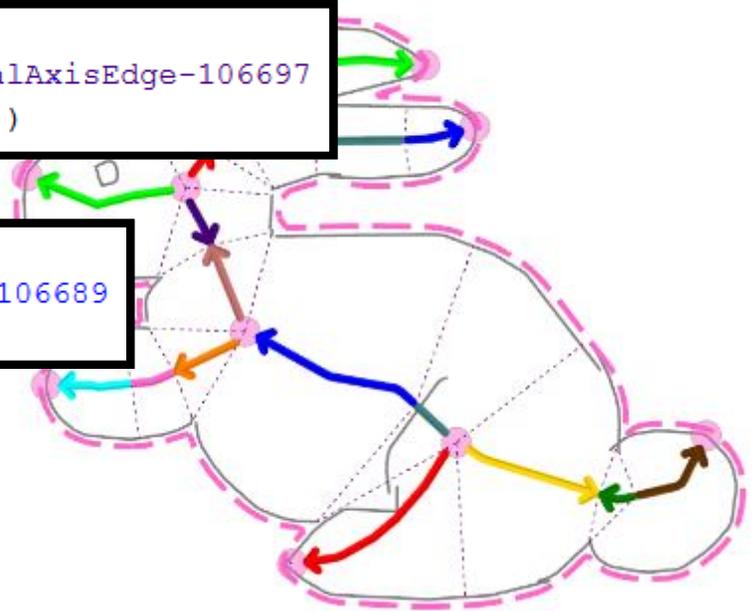
Obtuse/Right/Acute

Source/Sink

Relative to

```
(sinkConnection  
(elementsConnected MedialAxisEdge-106697  
MedialAxisEdge-106699))
```

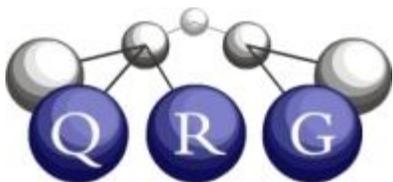
```
(stronglyDirectedConnection  
(directedConnection MedialAxisEdge-106689  
MedialAxisEdge-106699))
```



## Connection Radius Relations

Weakly/Strongly directed connection

Source/Sink connection



# MAT Representations

## Standard Edge Attributes

Edge

```
(acuteCorner  
  (elementsConnected MedialAxisEdge-106719  
    MedialAxisEdge-106713))  
  (clockwiseNeighborAtJunction MedialAxisEdge-106719  
    MedialAxisEdge-106713))
```

Source/Sink edge

Relative thickness

## Connection Radius Relations

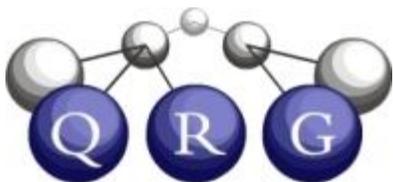
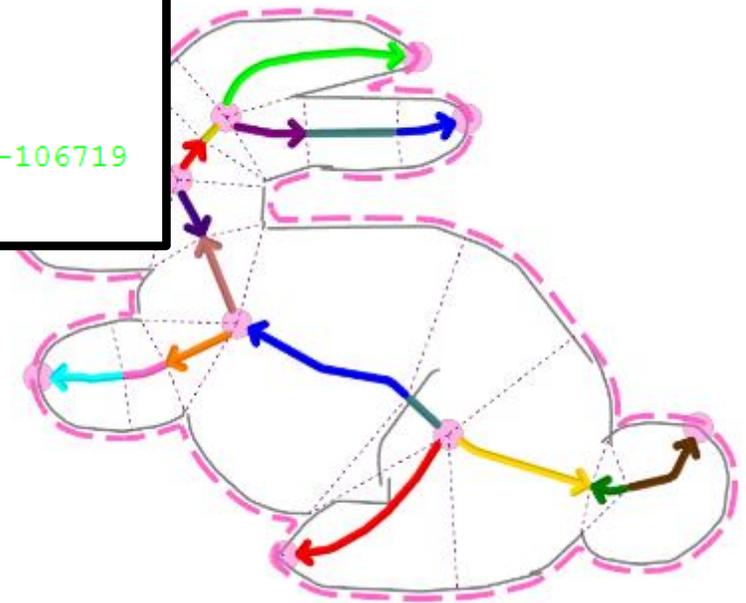
Weakly/Strongly directed connection

Source/Sink connection

## Connection Shape Relations

Obtuse/Right/Acute/Straight corner

Clockwise edge ordering



# Texture Detection

**Ising Model:** Given local grouping preferences (disagreement costs), finds globally optimal graph cut

Built on edge-cycle adjacency graph

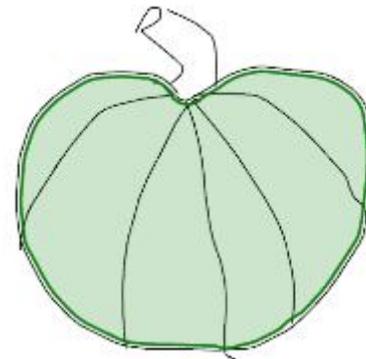
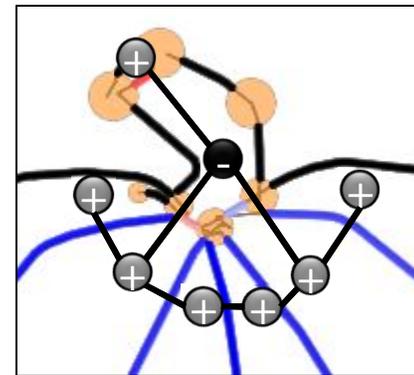
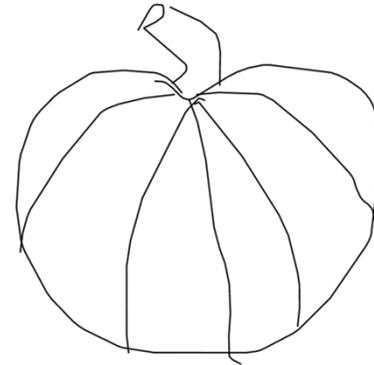
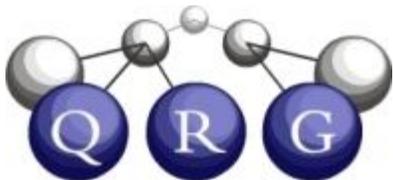
Disagreement costs

Real-valued (**affinity** or **anti-affinity**)

Computed from normalized perceptual similarities between edge-cycles along multiple dimensions

E.g. orientation, area, curvature

Details in (McLure et al. 2015)



# Texture Representations

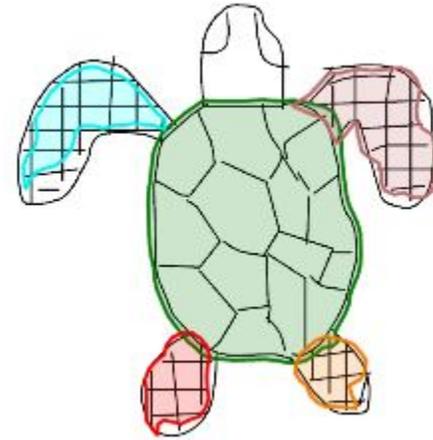
Textures are regions, which have perimeters and possibly holes

Perimeter and holes are edge-cycles

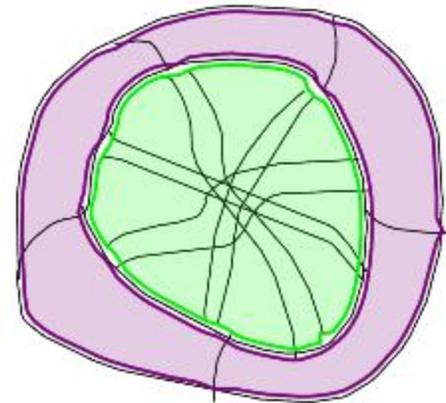
Perimeter and holes replace the grouped edge-cycles at the edge-cycle level of representation

Query predicate:

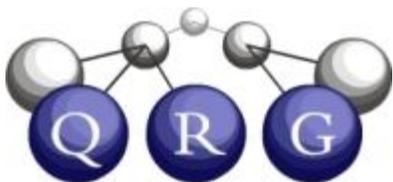
`edgeCycleRepresentationsFor-Textured`



~2500 facts → ~250 facts



~1000 facts → ~40 facts

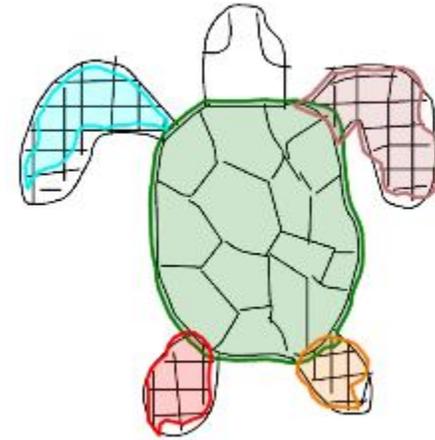


# Texture Representations

Texture region is reified

Tied to perimeter and holes with relations.

Attributes are assigned based on which dimensions of perceptual similarity correlate with the grouping

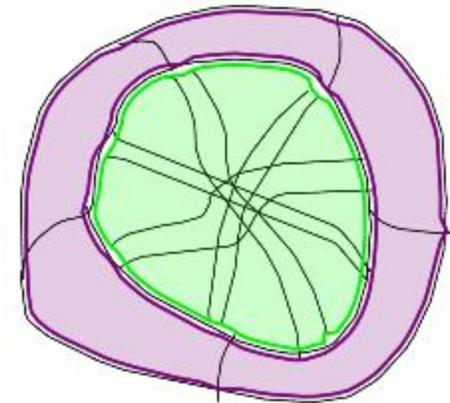


~2500 facts → ~250 facts

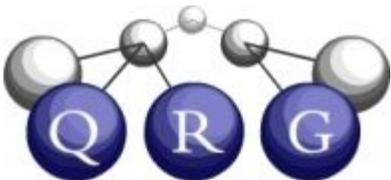
```
(hasHole EdgeCycle-Texture-3848276  
  (TextureHoleFn EdgeCycle-Texture-3848276 0))
```

```
(hasPerimeterEdgeCycle EdgeCycle-Texture-3848276  
  (PerimeterEdgeCycleFn ECO-3847663))
```

```
(isa EdgeCycle-Texture-3848276 PerceptualTexture-CycleSize)
```



~1000 facts → ~40 facts



# Querying for an analogy

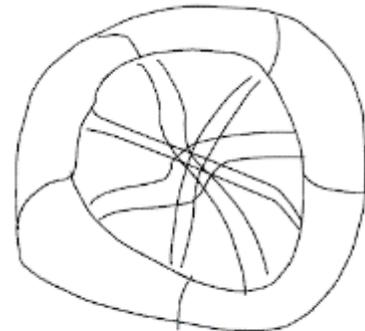
Query in the subsketch context

```
(ist-Information Bcase-3663043853  
  (matchBetween  
    (WMCASEFn (EdgeCycleFactsMtFn Tire1))  
    (WMCASEFn (EdgeCycleFactsMtFn Tire2))  
    (TheSet)  
    ?match))
```

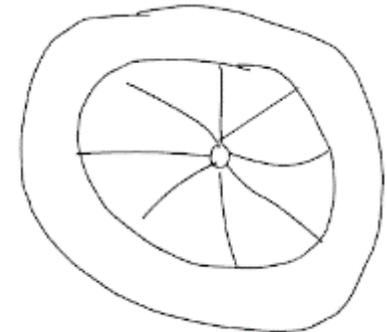
Two microtheories to compare

Match constraints (here, an empty set)

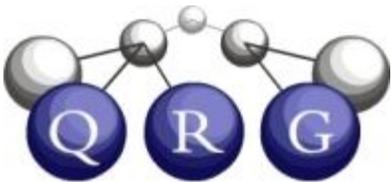
Gets bound to a match name



Tire1



Tire2



# The Analogy Browser

The screenshot displays the 'Query / WM Fact Edit' interface. The left sidebar contains a tree view of the case structure:

- Case-3663043853
  - Subsketch Subsketch 1
    - BCase-3663043853
      - Layer Positional
        - ObjectL-9497
      - Layer Voronoi
        - ObjectL-9496
      - Layer Layer 1
        - ObjectL-9134
      - Ex17989
        - Tire1
      - Ex17943
        - Tire2

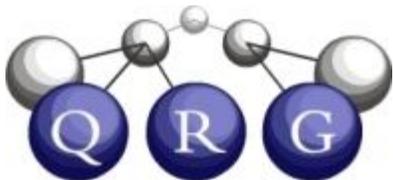
The main panel shows the query:

```
(matchBetween (WMCaseFn (EdgeCycleFactsMtFn Tire1))  
(WMCaseFn (EdgeCycleFactsMtFn Tire2)) (TheSet) ?match)  
action = ask  
context = BCase-3663043853; facts = all, env, infer
```

Below the query, the 'Answers:' section shows results for 'in BCase-3663043853':

```
? (matchBetween (WMCaseFn (EdgeCycleFactsMtFn Tire1))  
X (WMCaseFn (EdgeCycleFactsMtFn Tire2)) (TheSet)  
(MatcherFn 8 0))
```

A 'Return to Query/Edit Page' button is located below the results. At the bottom of the sidebar, a menu is open with 'Analogy' selected, highlighted by a blue callout box containing the text 'Access analogy browser'.



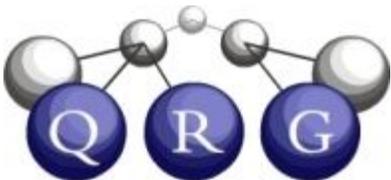
# The Analogy Browser

The screenshot displays the Analogy Browser interface. On the left is a sidebar with a tree view containing 'Case-3663043853', 'Subsketch Subsketch 1' (with sub-item 'BCase-3663043853'), 'Layer Positional' (with sub-item 'ObjectL-9497'), and 'Layer Voronoi' (with sub-item 'ObjectL-9496'). Below the sidebar is a menu with options: 'Clarify Glyph Relationships', 'Query / WM Fact Edit', 'KB Fact Edit', 'Analogy', 'Refresh Object List', 'Browse all WM', and 'Browse KB'. The main panel is titled 'Reasoner Analogy Source' and shows 'nuSketch Reasoner #23' and 'opencyc4 KB (HORCRUXWHOA)'. It features a 'Most Recent SME:' section with 'SME #8' and links for '[dehydrate]' and '[display analogy in CogSketch]'. Below this, it lists 'Base: (WMCaseFn (EdgeCycleFactsMtFn Tire1))' and 'Target: (WMCaseFn (EdgeCycleFactsMtFn Tire2))'. A 'Records Recorded in Reasoner Analogy Source:' section shows 'SME #8, version 0' with similar links and base/target information. A 'Cached DGroups:' section lists two entries: '(WMCaseFn (EdgeCycleFactsMtFn Tire2)) context = BCase-3663043853' and '(WMCaseFn (EdgeCycleFactsMtFn Tire1)) context = BCase-3663043853'. At the bottom of the main panel is a 'Clear Analogy Source' button.

Browse the match

- Similarity score
- Correspondences
- Candidate inferences

Visualize the correspondences



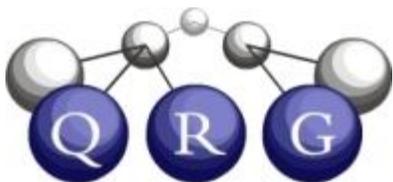
# The Analogy Display

Analogy Results

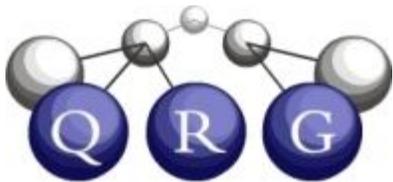
Base: (WMCaseFn) Mapping#104 (score = 0.124) Target: (WMCaseFn)

(PerimeterEdgeCycleFn ECO-3834760)	0.036	(PerimeterEdgeCycleFn ECO-3869282)
EdgeCycle-Texture-3868854	0.032	EdgeCycle-Texture-3869470
(PerimeterEdgeCycleFn (TextureHoleFn EdgeCycle-Texture-3868963 0))	0.028	(PerimeterEdgeCycleFn ECO-3869281)
(TextureHoleFn EdgeCycle-Texture-3868963 0)	0.008	ECO-3869281
ECO-3834760	0.004	(GlyphFn Tire2 User-Drawn-Sketch-Layer-9134)
EdgeCycle-Texture-3868963	0.004	ECO-3869282

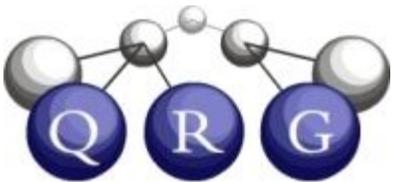
Show Raw Form? Browse SME



# Questions?

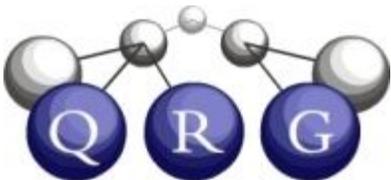


# Perceptual Sketchpad



# Perceptual Sketchpad Motivation

- Facility for experimenting with expressive representation of shapes
  - Decomposing glyphs
  - Modeling human shape comparisons
- Still experimental, hence separate subsystem
  - Not all CogSketch users need it
  - Much has now been integrated into educational worksheets

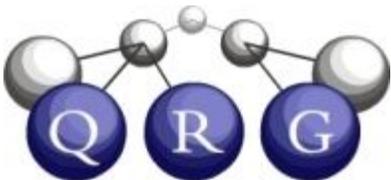


# Using the Perceptual Sketchpad

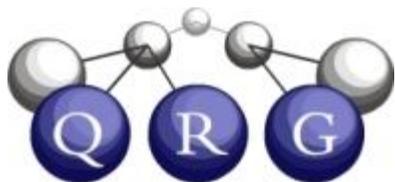
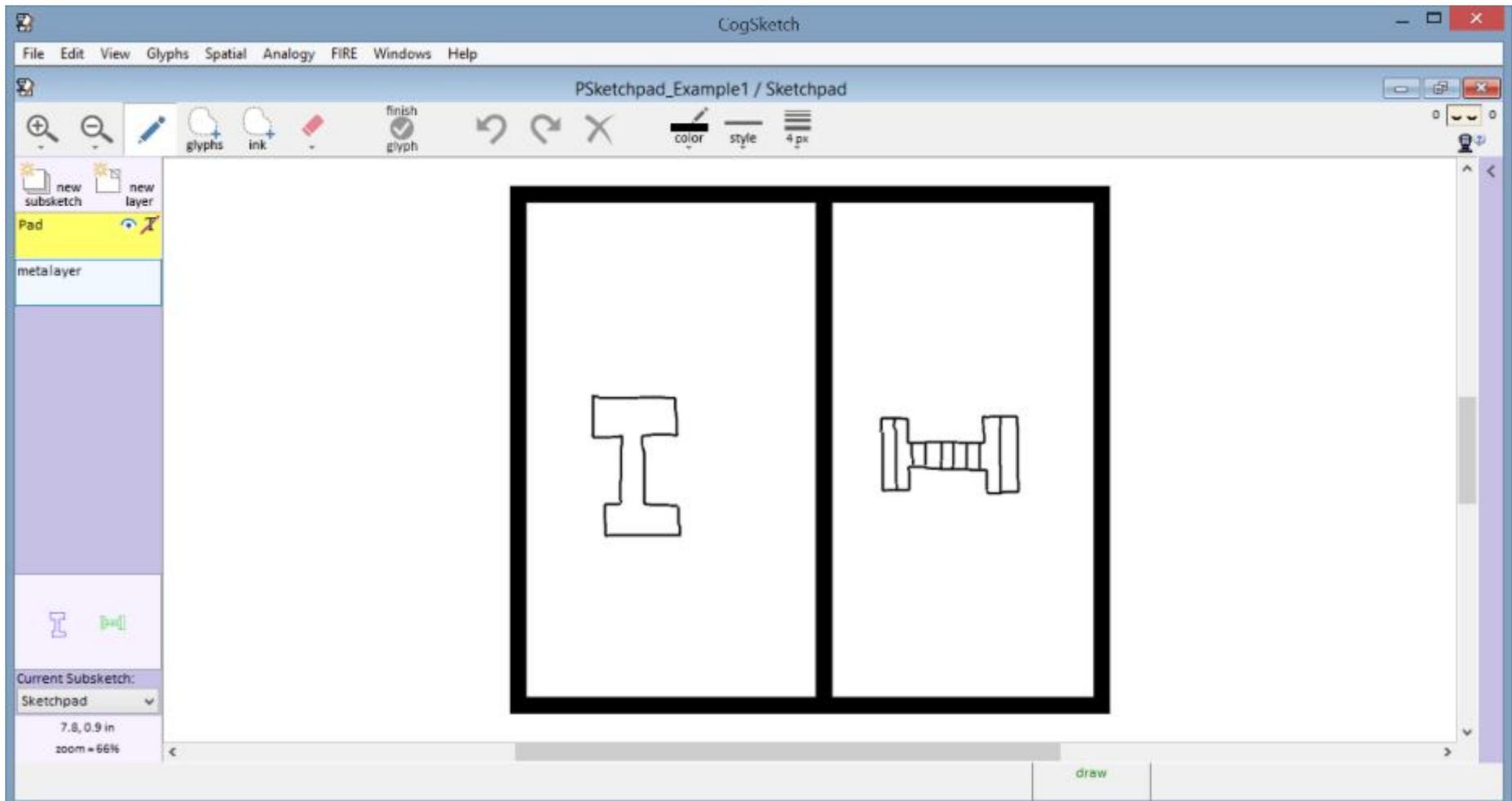
- CogSketch comes with a Perceptual Sketchpad demo
  - Choose “New Perceptual Sketchpad” from the File Menu

OR

- Open one of the examples from the sketches directory
  - PSketchpad\_Example1
  - PSketchpad\_Example2



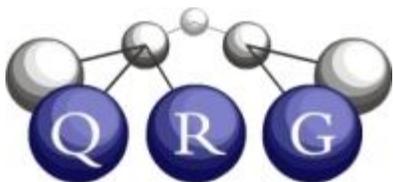
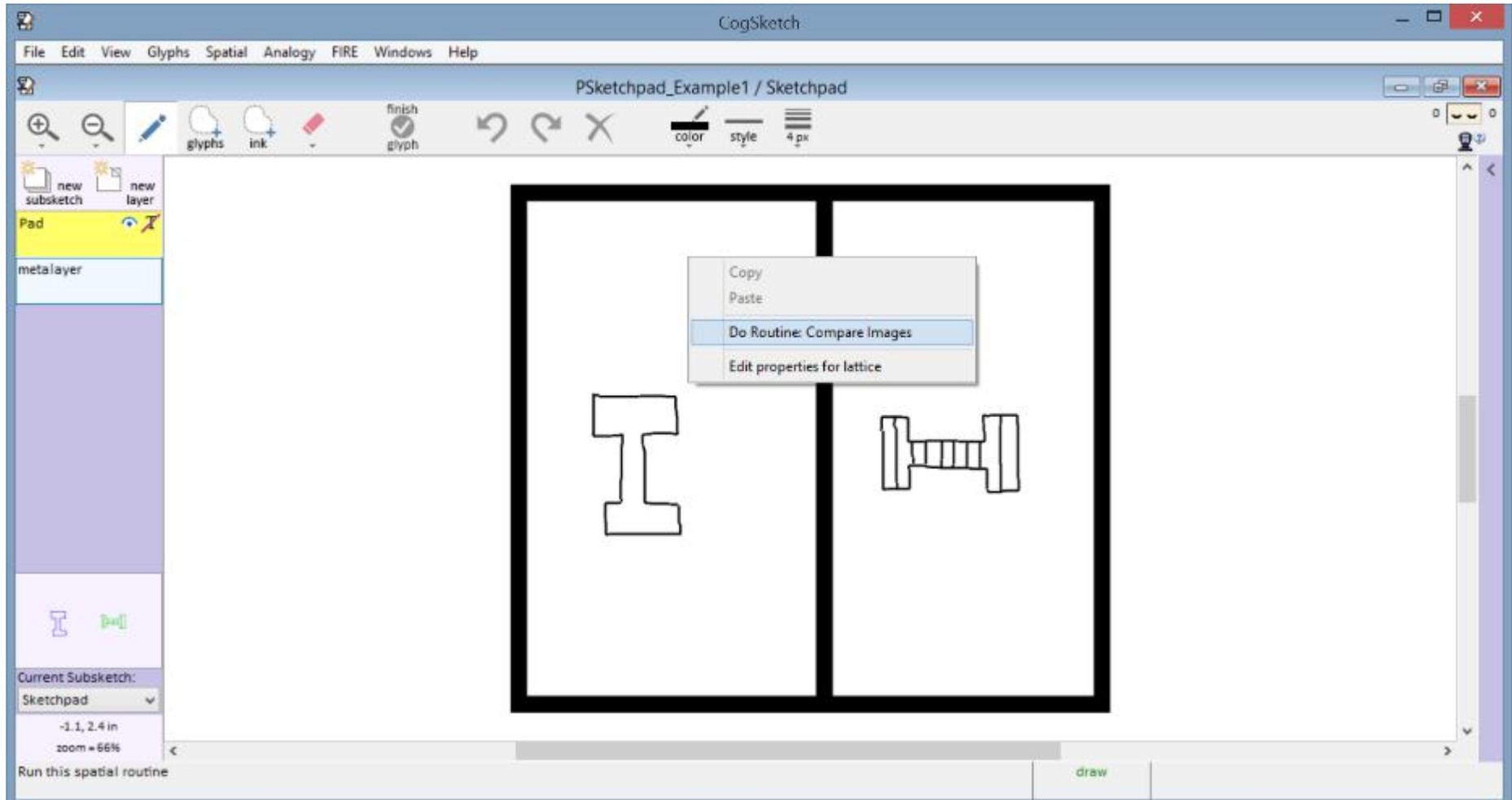
# Using the Perceptual Sketchpad



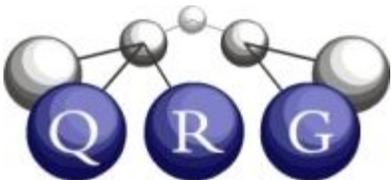
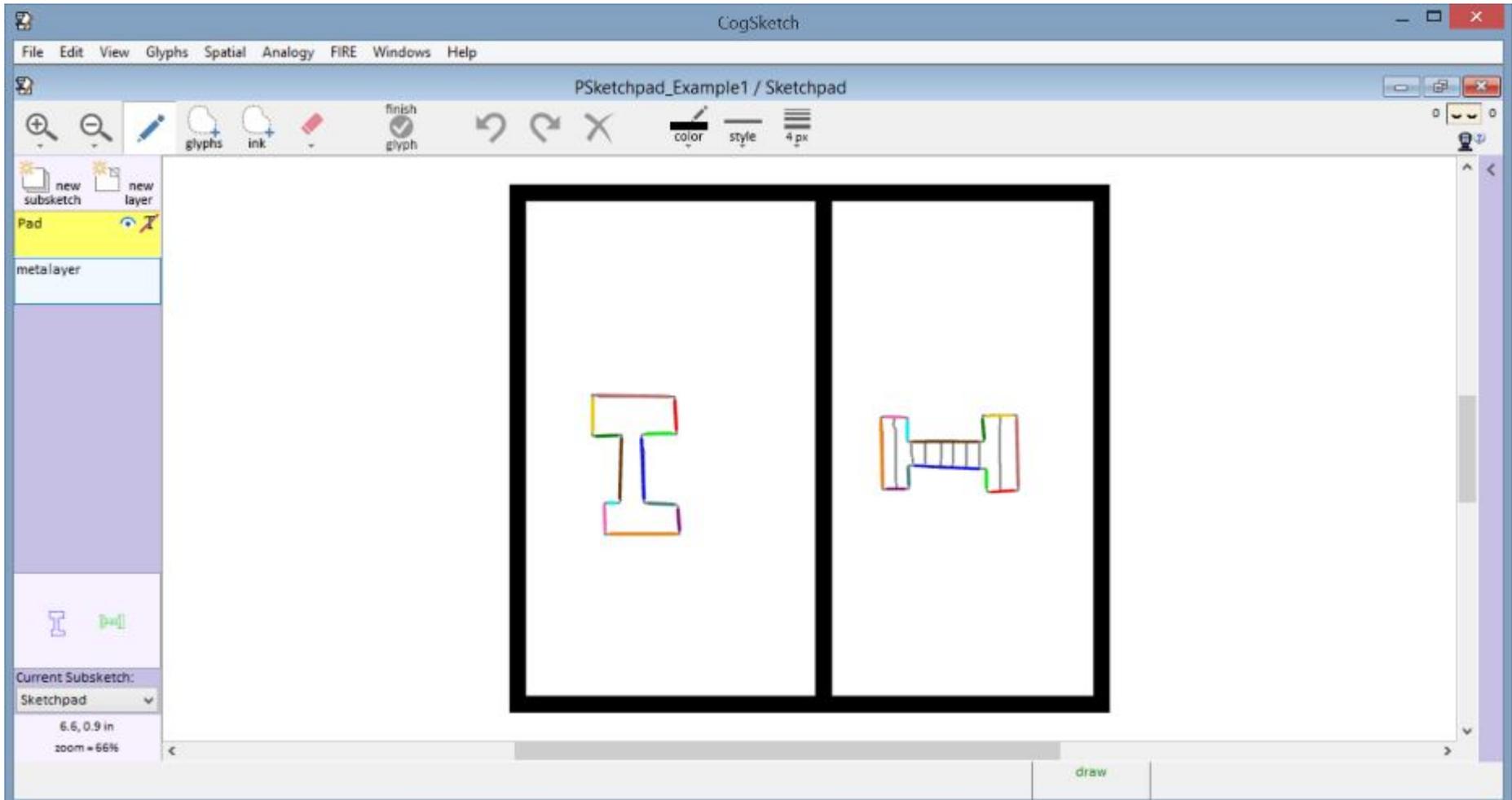
PSketchpad\_Example1



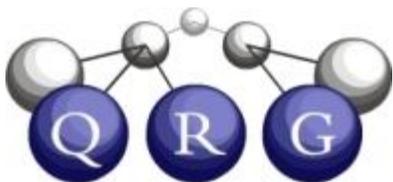
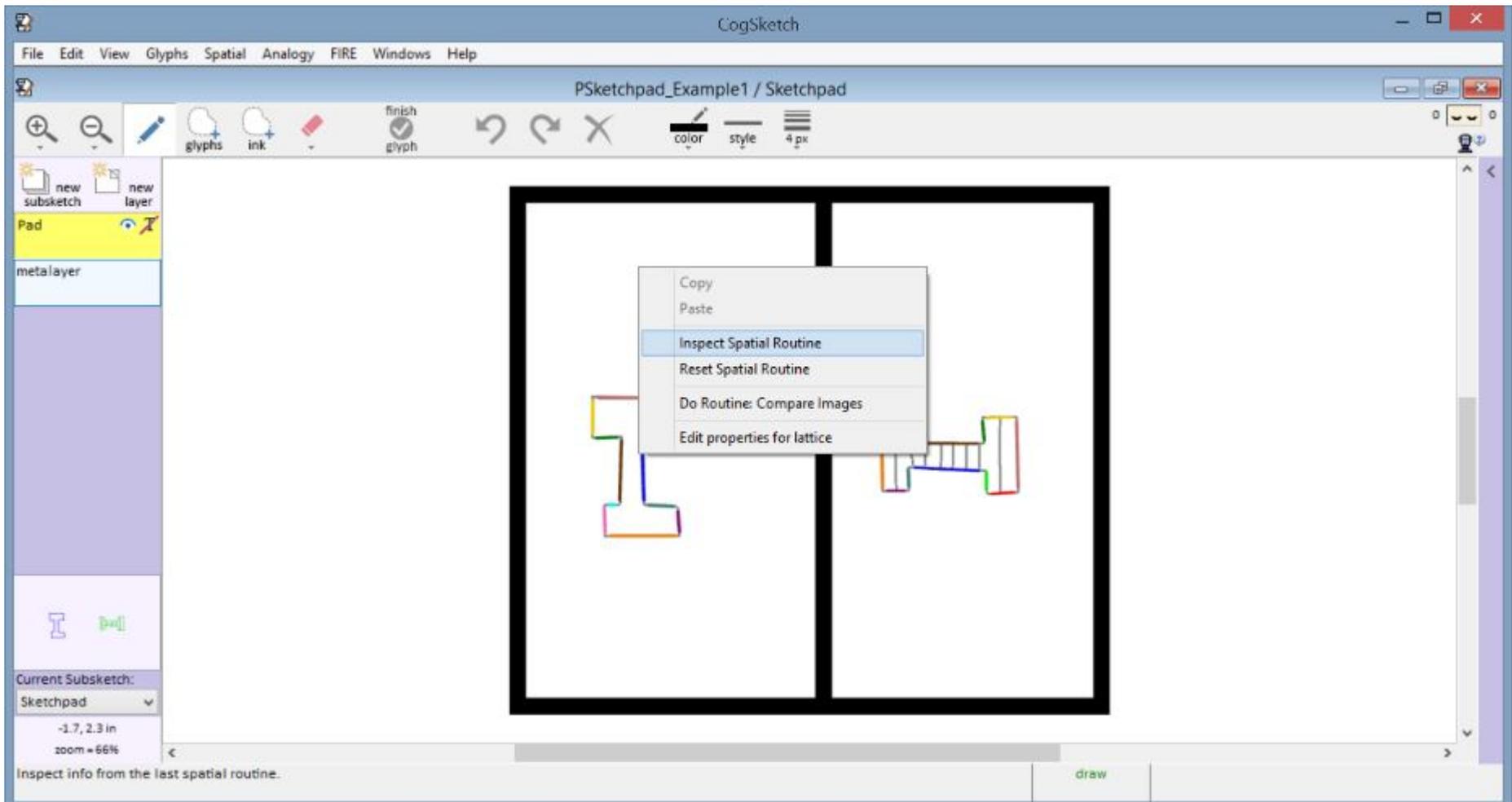
# Using the Perceptual Sketchpad



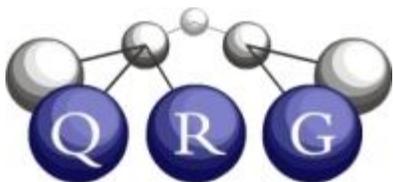
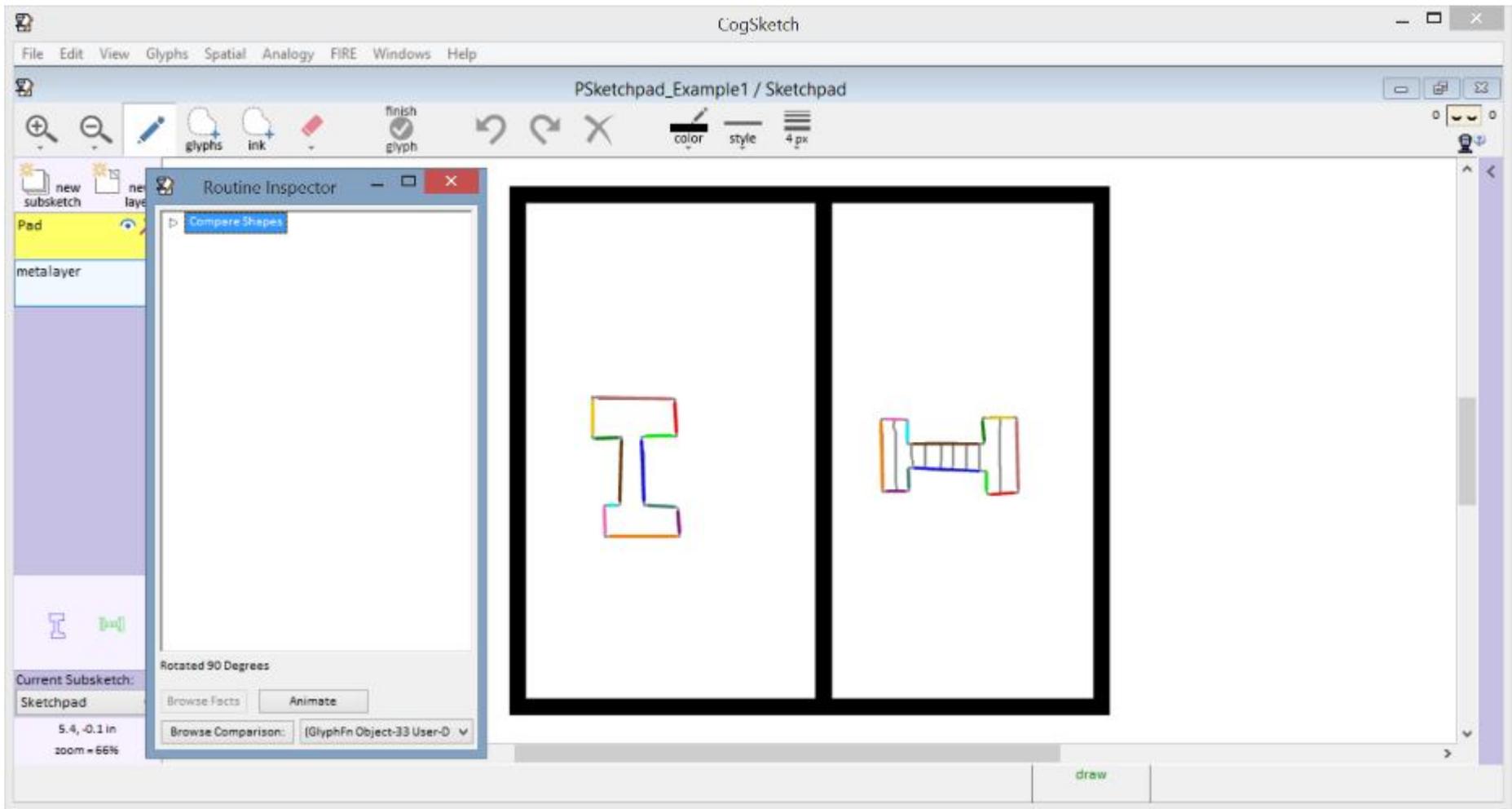
# Using the Perceptual Sketchpad



# Using the Perceptual Sketchpad

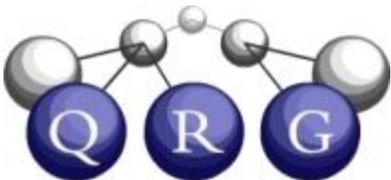


# Using the Perceptual Sketchpad

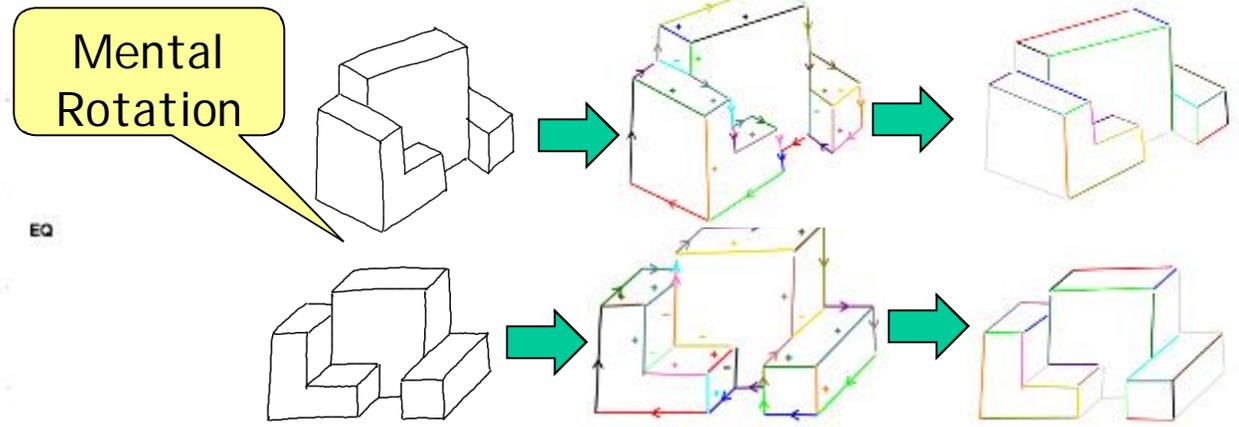
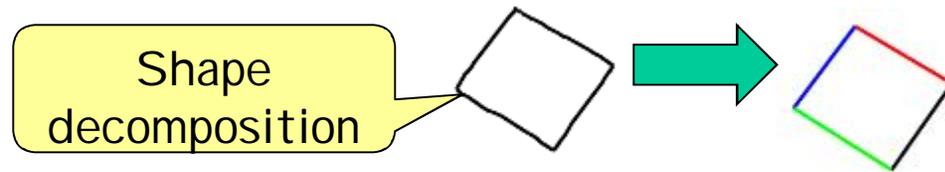
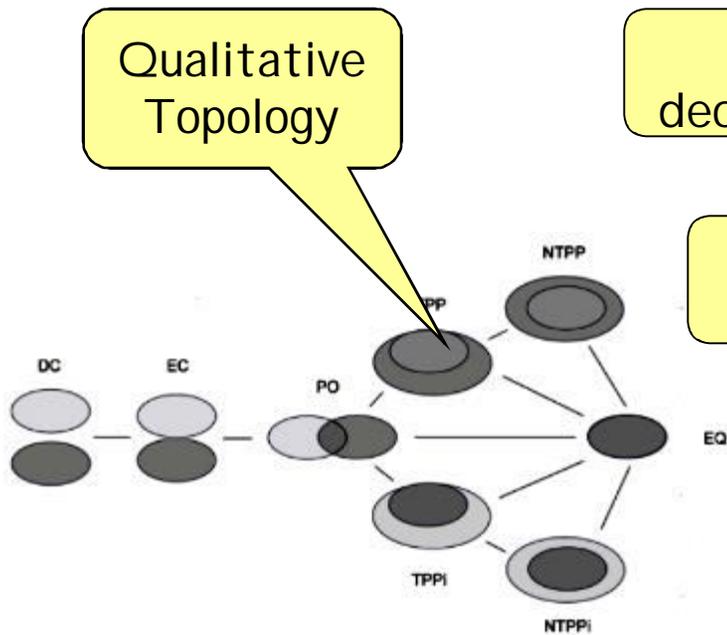
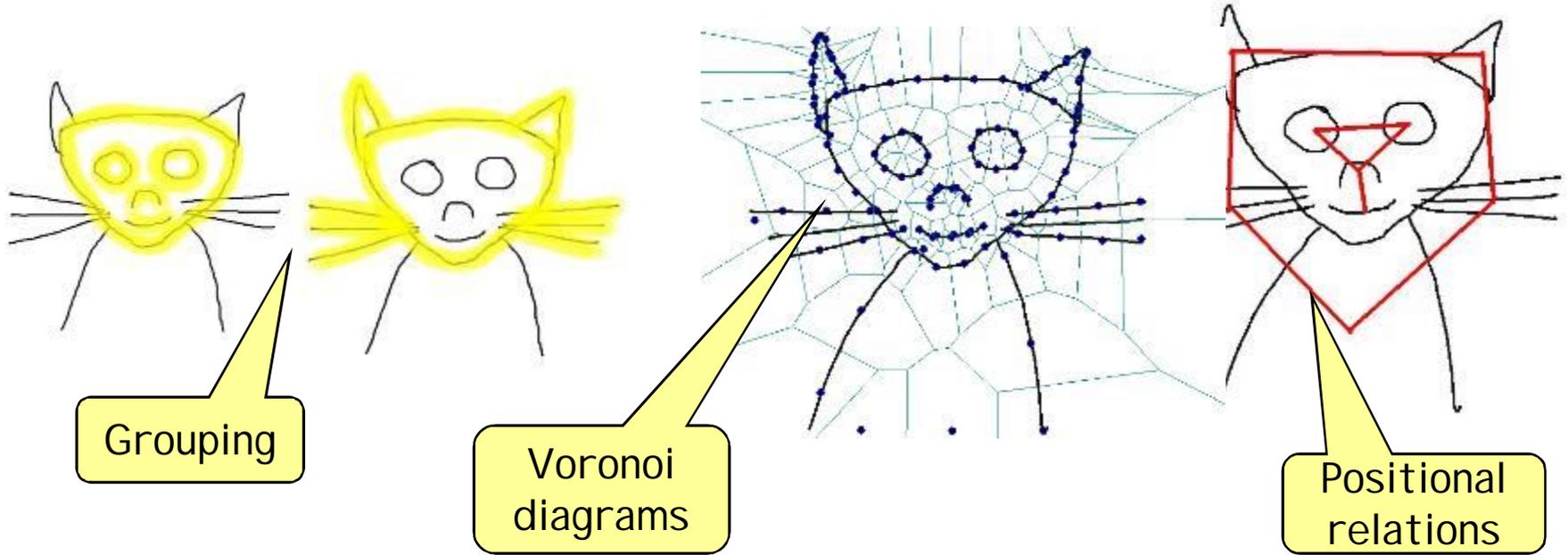


# Using the Perceptual Sketchpad

- If there is one glyph in each entry
  - Edge-level representations will be used
- If there are multiple glyphs
  - Glyph-level representations will be used
- Elements will be color-coded to indicate correspondences
  - Right-click and choose “Reset Spatial Routine” to remove colors



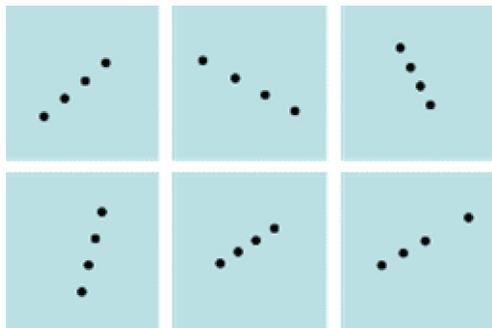
# Some CogSketch spatial computations



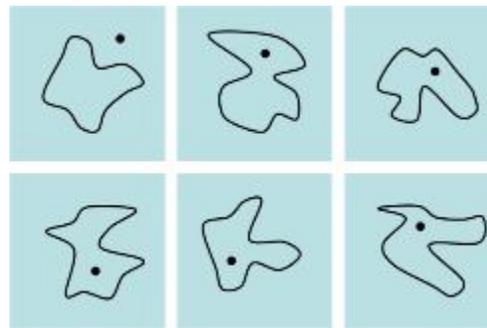
# Hierarchical Representations in Psychology

Images can be represented at different levels in a spatial hierarchy

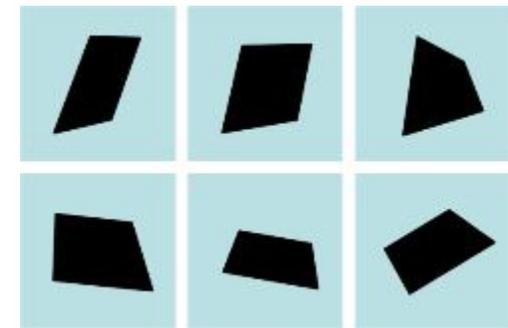
Spatial reasoning requires identifying the appropriate level



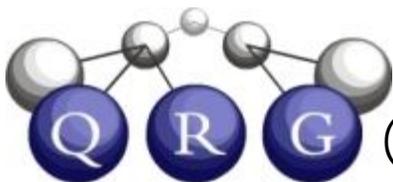
Groups



Objects



Edges



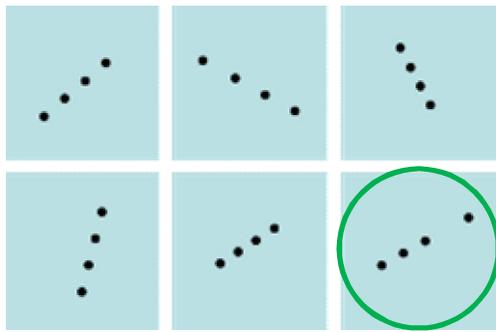
(Marr & Nishihara, 1978; Palmer, 1977; Hochstein & Ahissar, 2002)



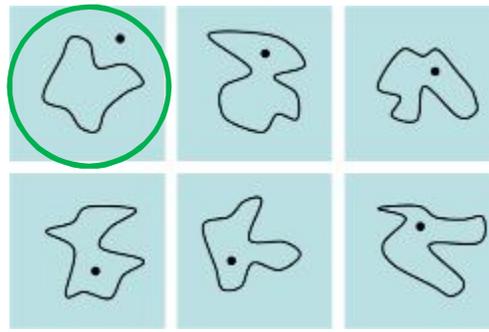
# Hierarchical Representations in Psychology

Images can be represented at different levels in a spatial hierarchy

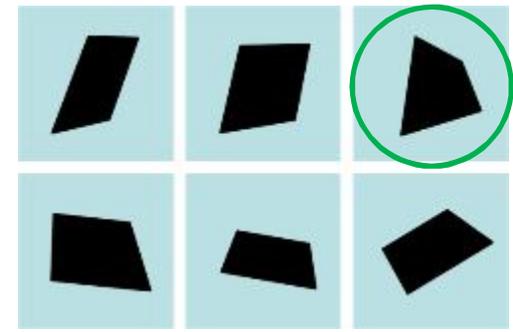
Spatial reasoning requires identifying the appropriate level



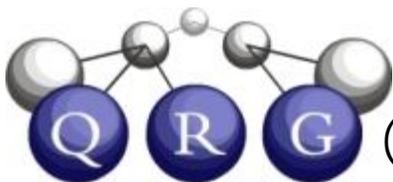
Groups



Objects



Edges



(Marr & Nishihara, 1978; Palmer, 1977; Hochstein & Ahissar, 2002)

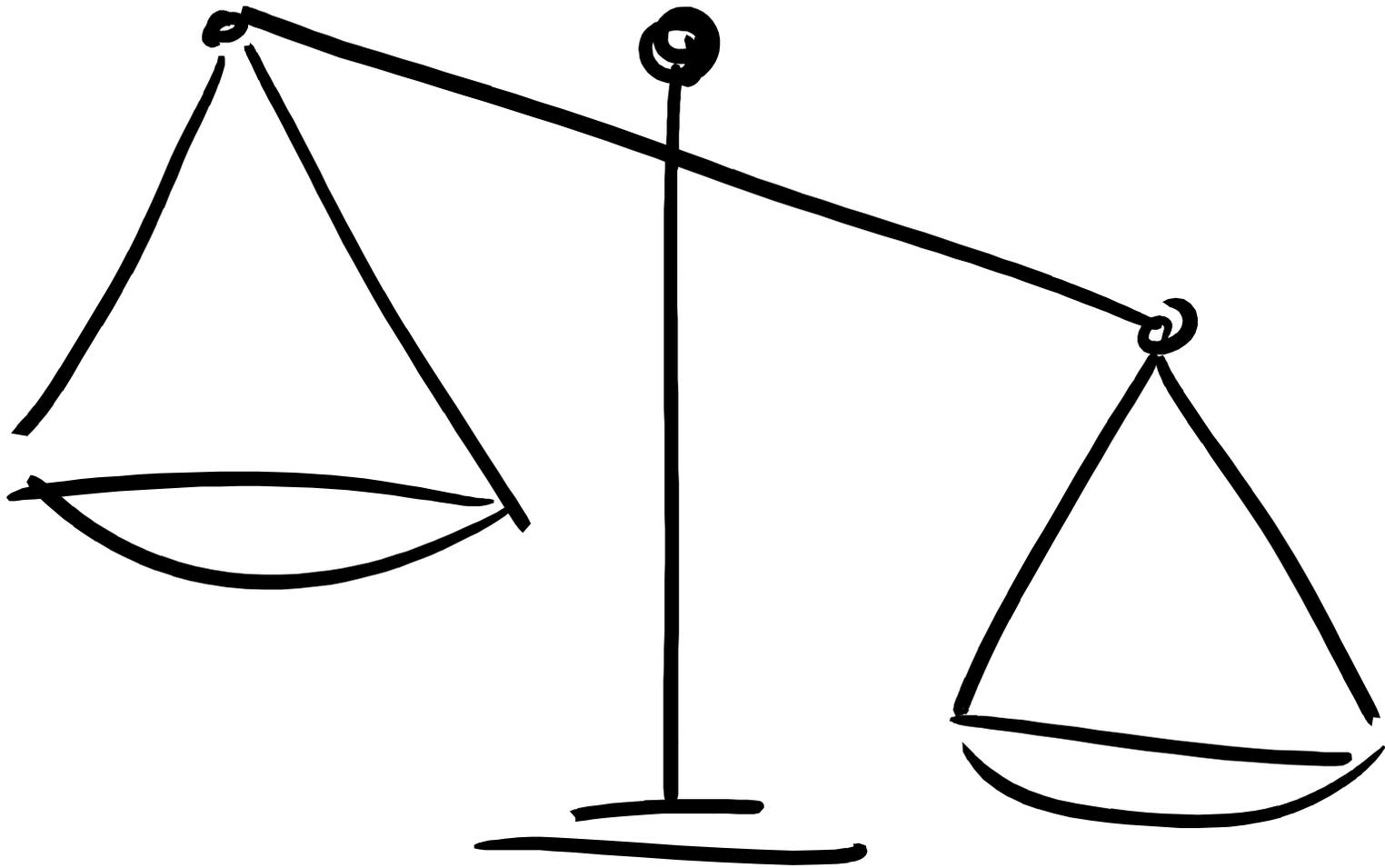


# CogSketch and Education

Tutorial

AAAI 2016

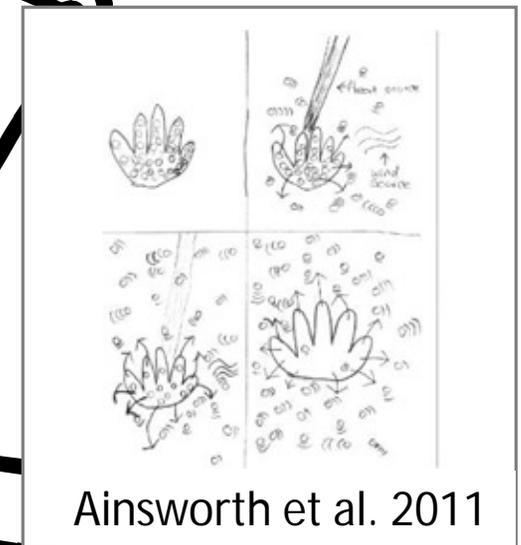
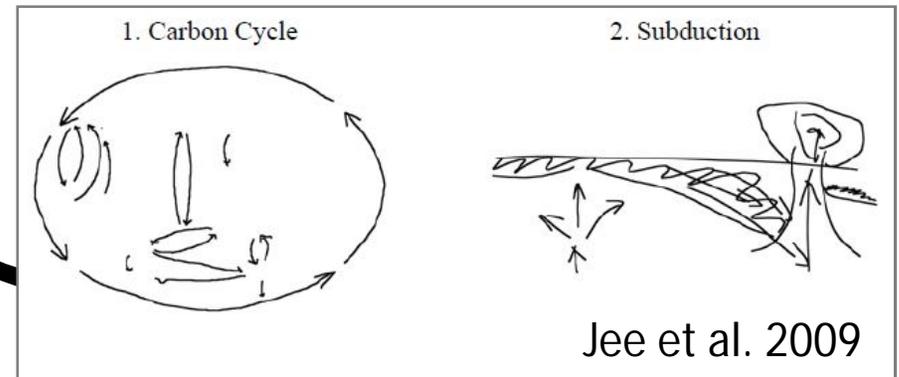
# Sketching for Learning in Science



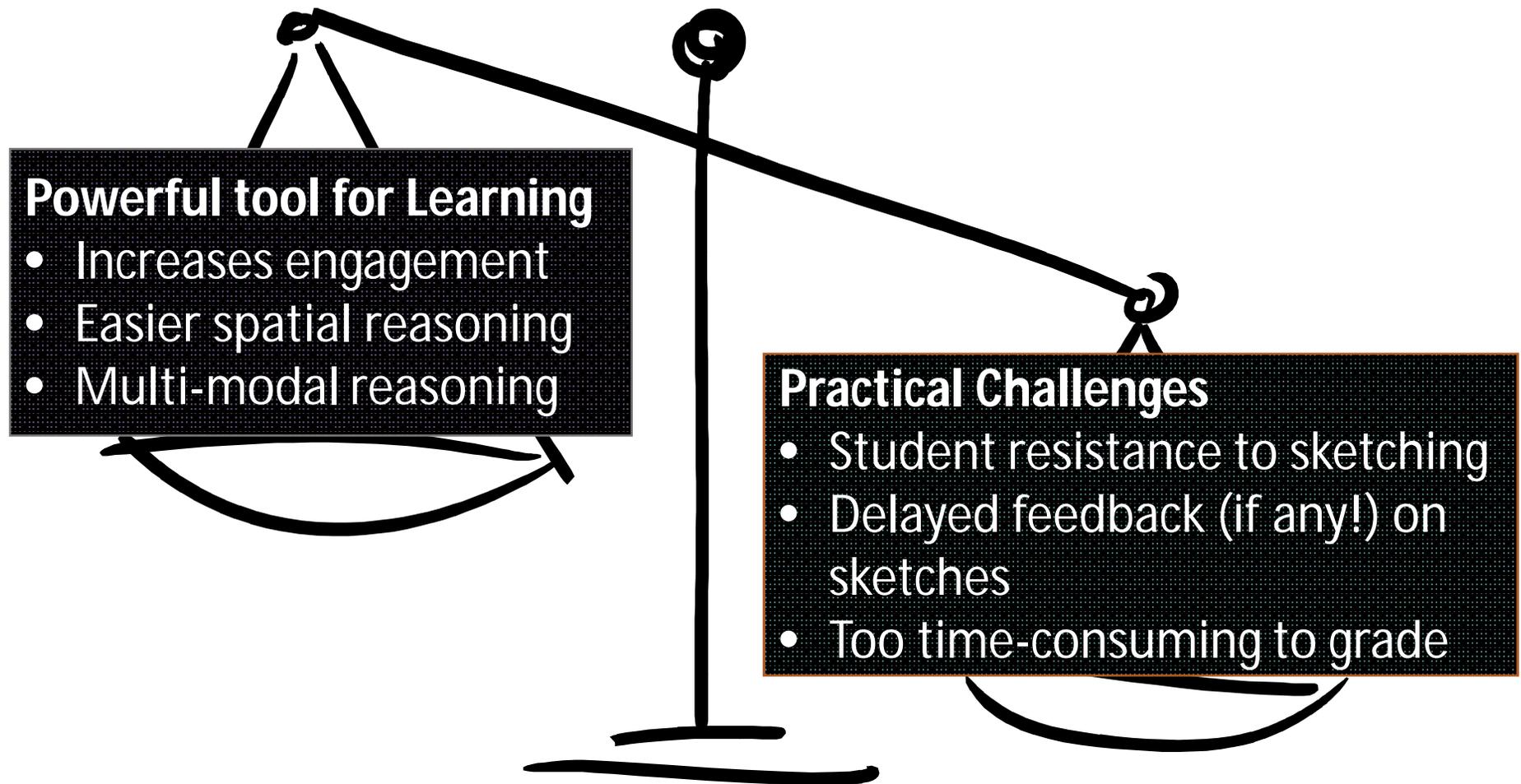
# Sketching for Learning in Science

## Powerful tool for Learning

- Increases engagement
- Easier spatial reasoning
- Multi-modal reasoning



# Sketching for Learning in Science



# Benefits of Intelligent Tutoring Systems

*Each challenge can be at least partially addressed by a sketch-based intelligent tutoring system (SBITS)*

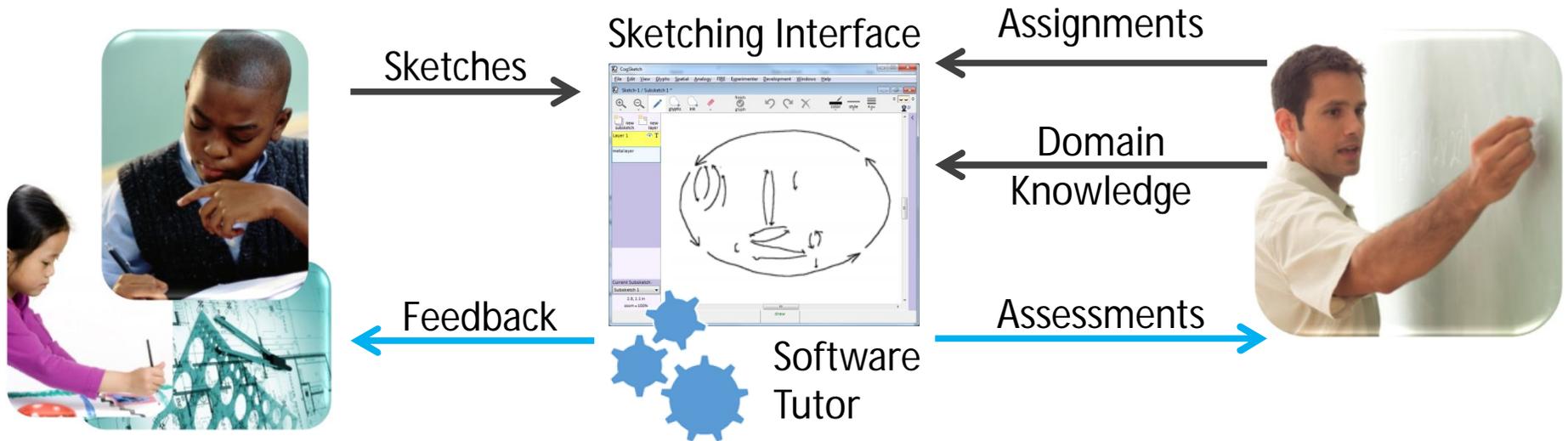
## **Practical Challenges of Sketching**

- Student resistance to sketching
- Delayed feedback (if any!) on sketches
- Too time-consuming to grade

- Safe environment for students to practice, make mistakes, learn
- Timely, on-the-spot feedback
- Automated assessment

# CogSketch as a Platform for sketch-based educational software

*Eventually, like a calculator, but with automated feedback and assessment*



Our vision: sketch understanding software for helping students learn could be widely available within 3-5 years.

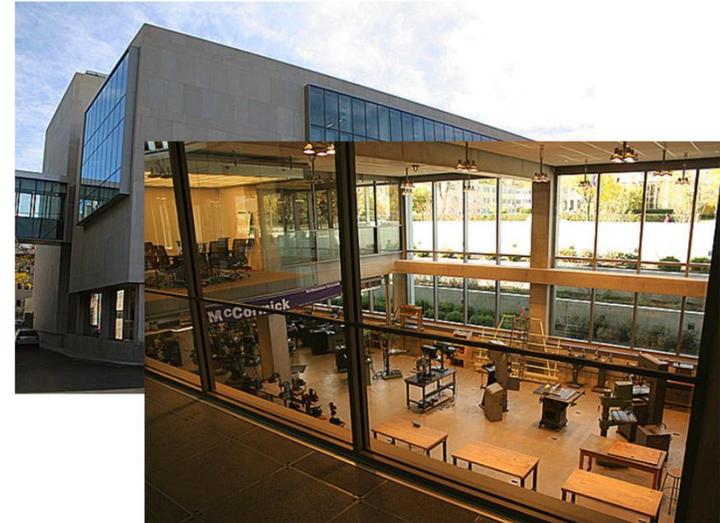
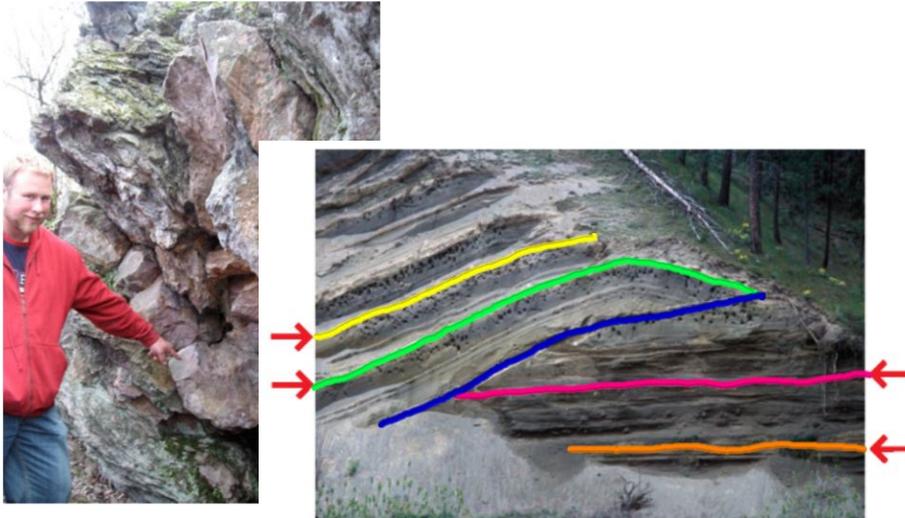
# Sketch-based Intelligent Tutoring: Technical Challenges

- Sketch Understanding
  - Visual Processing
  - Qualitative Spatial Reasoning
  - Visual-conceptual reasoning
  - Domain knowledge
- Intelligent tutoring
  - Natural interaction
  - Teaching knowledge
  - Student modeling
  - Domain knowledge

via CogSketch

Two approaches:  
Domain-specific  
Domain-general

# Current CogSketch Education Projects



---

## Sketch Worksheets

### Problem

Many STEM fields require learning **spatial layouts and terminology**, but students do not get timely feedback on sketching exercises.

### Idea

Develop software to coach students on sketching exercises

### Domain

Potentially any

### People

Maria Chang, Jeff Usher

---

## Design Coach

Students have trouble using **sketches to communicate their ideas**.

Develop software to coach students on their design sketches and explanations

Engineering Design

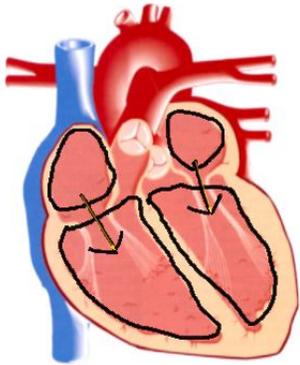
Jon Wetzel

---

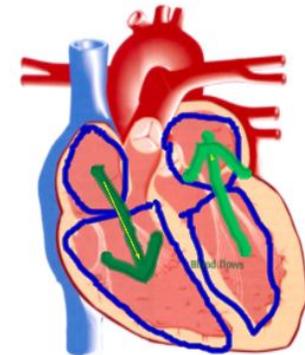
# Sketch Worksheets: domain-general, model-based tutoring

(Yin et al. 2010)

Teacher's Solution



Student's Solution



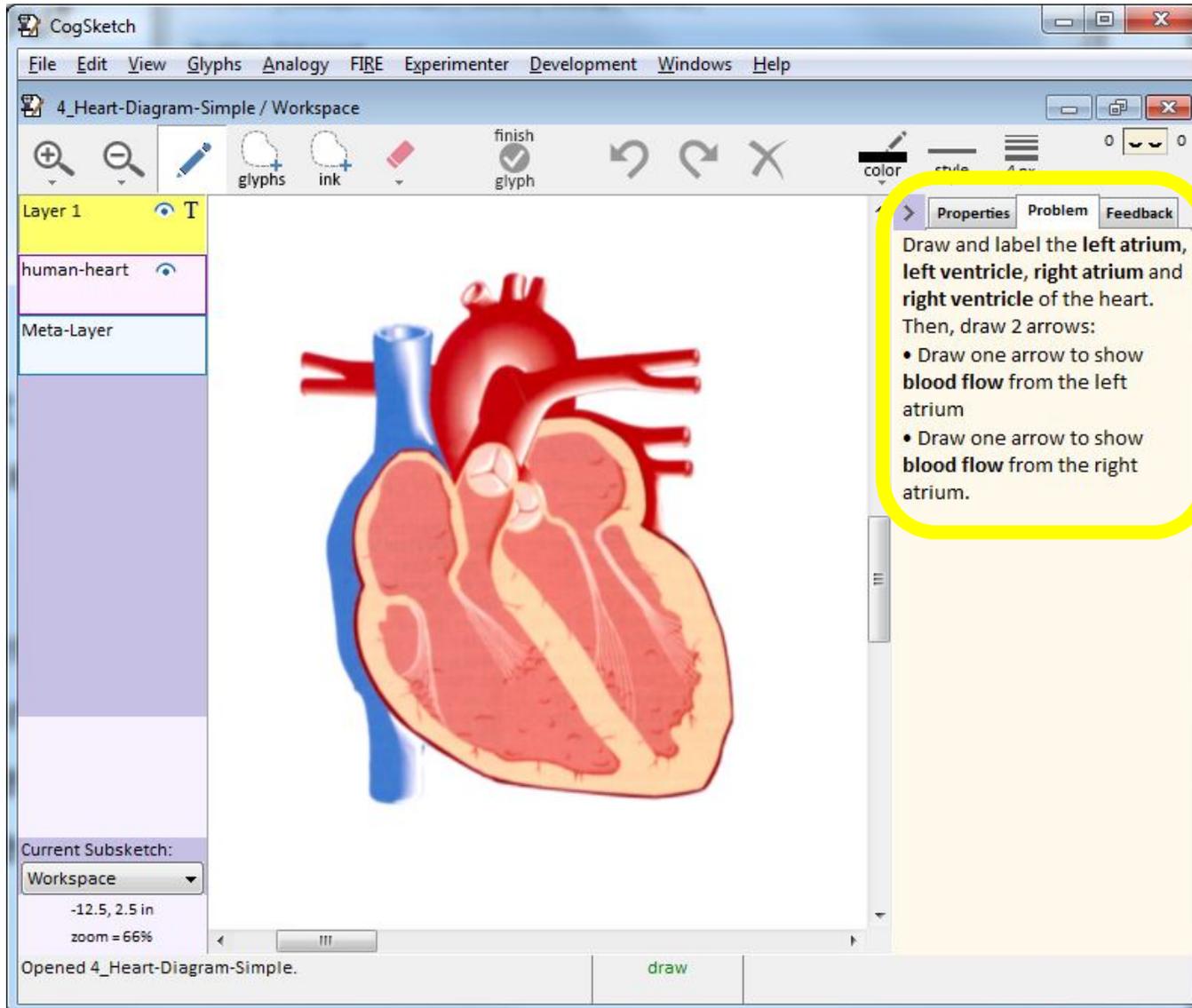
Structure-Mapping Engine

Differences between sketches +  
Pre-determined correctness criteria

Feedback to Student:  
*"The left atrium pumps blood out through a valve.  
Where does that blood go?"*

- Mimic traditional paper and pencil worksheets
- Provides on-demand feedback to students
- Platform for instructors to create *their own* worksheets

# Worksheet Example – Instructions



The screenshot displays the CogSketch software interface. The main workspace contains a 3D anatomical diagram of a human heart. On the right side, a task instruction panel is highlighted with a yellow border. The panel has three tabs: 'Properties', 'Problem', and 'Feedback'. The 'Problem' tab is selected, showing the following instructions:

Draw and label the **left atrium**, **left ventricle**, **right atrium** and **right ventricle** of the heart. Then, draw 2 arrows:

- Draw one arrow to show **blood flow** from the left atrium
- Draw one arrow to show **blood flow** from the right atrium.

The interface also shows a menu bar (File, Edit, View, Glyphs, Analogy, FIRE, Experimenter, Development, Windows, Help), a toolbar with various drawing tools (pen, eraser, lasso, etc.), and a layer panel on the left with 'Layer 1' selected. The status bar at the bottom indicates 'Opened 4\_Heart-Diagram-Simple.' and a 'draw' button.

# Worksheet Example – Draw, Finish Glyph

CogSketch

File Edit View Glyphs Analogy FIRE Experimenter Development Windows Help

4\_Heart-Diagram-Simple / Workspace

finish glyph

Layer 1

human-heart

Meta-Layer

Current Subsketch: Workspace

-15.5, -0.2 in  
zoom = 66%

Draw new ink on the current layer.

draw

Properties Problem Feedback

Draw and label the **left atrium**, **left ventricle**, **right atrium** and **right ventricle** of the heart. Then, draw 2 arrows:

- Draw one arrow to show **blood flow** from the left atrium
- Draw one arrow to show **blood flow** from the right atrium.

# Worksheet Example – Label Glyph

The screenshot displays the CogSketch software interface. The main workspace shows a 3D anatomical diagram of a human heart. A red circle, representing a label glyph, is drawn around the left atrium. The text "Left atrium" is positioned below the circle. The software's interface includes a menu bar (File, Edit, View, Glyphs, Analogy, FIRE, Experimenter, Development, Windows, Help), a toolbar with various drawing tools (zoom, ink, glyphs, finish glyph, undo, redo, delete, color, style, 10 px), and a right-hand panel with tabs for Properties, Problem, and Feedback. The Properties panel is highlighted with a yellow box and contains the following information:

- Glyph Type:** entity
- What is this?**
  - Human heart
  - Human heart chamber
  - Internal organ
  - Left atrium
  - Left ventricle
  - Lung
  - Right atrium
  - Right ventricle
- Name:** Left atrium
- Ink Properties:** color, style, 10 px

At the bottom of the interface, there is a status bar with the text "Choose what kind of thing your drawing represents." and a "draw" button.

# Worksheet Example – Request Feedback

The screenshot displays the CogSketch software interface. The main workspace shows a 3D anatomical model of a human heart with a 2D cross-section overlaid. The cross-section is outlined in purple and labeled with "Left atrium", "Right atrium", "Right ventricle", and "Left ventricle". The software's menu bar includes "File", "Edit", "View", "Glyphs", "Analogy", "FIRE", "Experimenter", "Development", "Windows", and "Help". The toolbar contains icons for zooming, drawing, and editing. A sidebar on the left shows a layer stack with "Layer 1" selected. A feedback panel on the right, highlighted with a yellow border, displays "Your Progress" with an "Update" button and several progress indicators:

- All Glyphs have Labels:  (green bar)
- Required Glyphs:  (green bar)
- Atria:  (yellow bar)
- Blood Flow Relations:  (yellow bar)
- Ventricles:  (yellow bar)
- Overall Sketch:  (yellow bar)

At the bottom of the interface, there is a "Current Subsketch:" dropdown menu set to "Workspace", a zoom level of "66%", and a "select entire glyphs" button.

# Worksheet Example – Drill down

You might have left and right mixed up in your sketch. Here's a hint: the left atrium isn't on the left-hand side from your perspective. The same goes for the left ventricle.

The screenshot shows the CogSketch software interface. The main window displays a heart diagram with four chambers outlined in different colors: the left atrium in green, the right atrium in red, the left ventricle in purple, and the right ventricle in orange. Labels for 'Left atrium', 'Right atrium', 'Right ventricle', and 'Left ventricle' are visible. The software's menu bar includes 'File', 'Edit', 'View', 'Glyphs', 'Analogy', 'FIRE', 'Experimenter', 'Development', 'Windows', and 'Help'. The toolbar contains various drawing tools like 'glyphs', 'ink', 'finish glyph', 'color', 'style', and '10px'. On the right side, there is a 'Properties' panel with sections for 'Your Progress', 'Atria', 'Blood Flow Relations', and 'Ventricles'. A yellow box highlights a feedback message in the 'Atria' section: 'You might have left and right mixed up in your sketch. Here's a hint: the left atrium isn't on the left-hand side from your perspective. The same goes for the left ventricle.'

# Worksheet Example – Revision

The screenshot displays the CogSketch software interface. The main workspace shows a 3D anatomical diagram of a human heart with several parts highlighted in blue and labeled: "Right atrium", "Left atrium", "Right ventricle", and "Left ventricle". Yellow curved arrows indicate a clockwise flow path starting from the top left, moving through the atria and ventricles. The interface includes a menu bar (File, Edit, View, Glyphs, Analogy, FIRE, Experimenter, Development, Windows, Help), a toolbar with various drawing tools (glyphs, ink, finish glyph, color, style, 10px), and a left sidebar with layers (Layer 1, human-heart, Meta-Layer) and a "Current Subsketch" dropdown set to "Workspace".

On the right side, a "Properties" panel titled "Your Progress" shows the following status:

- Update: Update
- All Glyphs have Labels:
- Required Glyphs:
- Atria:
- Blood Flow Relations:
- Ventricles:
- Overall Sketch:

At the bottom of the interface, there is a "draw" button and a status bar that reads "Updates the feedback and progress meters."

# Worksheet Example – Drill Down

The screenshot displays the CogSketch software interface. The main workspace shows a 3D anatomical diagram of a human heart with four chambers labeled: Right atrium, Left atrium, Right ventricle, and Left ventricle. The diagram is overlaid with blue and purple outlines. The interface includes a menu bar (File, Edit, View, Glyphs, Analogy, FIRE, Experimenter, Development, Windows, Help), a toolbar with various drawing tools (glyphs, ink, finish glyph, color, style, 10px), and a left sidebar with layers (Layer 1, human-heart, Meta-Layer) and a 'Current Subsketch' dropdown set to 'Workspace'. The bottom status bar shows coordinates (-10.5, -2.2 in) and a zoom level of 66%. On the right, a 'Properties' panel is open, showing 'Your Progress' with an 'Update' button and several progress indicators. The 'Blood Flow Relations' section is highlighted with a yellow box and contains the following text:

**Blood Flow Relations**

- The right atrium pumps blood out through a valve. Where does that blood go?
- The left atrium pumps blood out through a valve. Where does that blood go?

# Worksheet Example – Blood flow arrows

The screenshot displays the CogSketch software window. The title bar reads "CogSketch". The menu bar includes "File", "Edit", "View", "Glyphs", "Analogy", "FIRE", "Experimenter", "Development", "Windows", and "Help". The workspace title is "4\_Heart-Diagram-Simple / Workspace \*".

The toolbar contains icons for zooming, drawing, erasing, and finishing glyphs. The layer list on the left shows "Layer 1" (selected), "human-heart", and "Meta-Layer".

The central workspace shows a 3D heart diagram with labels: "Right atrium", "Left atrium", "Right ventricle", and "Left ventricle". A black arrow points from the right atrium towards the right ventricle.

The right-hand panel shows a "Your Progress" section with an "Update" button and several progress indicators:

- All Glyphs have Labels:
- Required Glyphs:
- Atria:
- Blood Flow Relations: 
  - The right atrium pumps blood out through a valve. Where does that blood go?
  - The left atrium pumps blood out through a valve. Where does that blood go?
- Ventricles:
- Overall Sketch:

At the bottom of the window, there is a "select entire glyphs" button.

# Worksheet Example – Blood flow arrows

The screenshot displays the CogSketch software interface. The main workspace shows a 3D anatomical diagram of a human heart with four chambers labeled: Right atrium, Left atrium, Right ventricle, and Left ventricle. A red arrow labeled "Blood flows" points from the Right atrium to the Right ventricle. The interface includes a menu bar (File, Edit, View, Glyphs, Analogy, FIRE, Experiment, Development, Windows, Help), a toolbar with various drawing tools, and a left sidebar with layers (Layer 1, human-heart, Meta-Layer) and a "Current Subsketch" section. On the right, a "Properties" panel is highlighted with a yellow border, showing the "Glyph Type" set to "relation". Under "What is this?", the "Blood flows" option is checked, with a description: "Indicates that blood flows from one place to another. For example, (bloodFlowsFromAndTo RightVentricle Lungs) means that blood flows from the right ventricle to the lungs." Below this, another checked option reads "Blood flows from Right atrium to Right ventricle." The "Name" field contains "Blood flows". The "Ink Properties" section shows a color selection tool, a style selection tool, and a line width of 10 px. At the bottom, a status bar contains the text "Choose what kind of thing your drawing represents." and a button labeled "select entire glyphs".

# Worksheet Example – Completed

The screenshot displays the CogSketch software interface. The main workspace shows a 3D anatomical model of a human heart with a hand-drawn purple outline and black arrows indicating blood flow. Labels include "Right atrium", "Left atrium", "Right ventricle", "Left ventricle", and "Blood flows". The interface includes a menu bar (File, Edit, View, Glyphs, Analogy, FIRE, Experimenter, Development, Windows, Help), a toolbar with icons for zooming, drawing, and editing, and a layer panel on the left with "Layer 1", "human-heart", and "Meta-Layer". A progress panel on the right, titled "Your Progress", shows a list of tasks with green progress bars and checkmarks: "All Glyphs have Labels", "Required Glyphs", "Atria", "Blood Flow Relations", "Ventricles", and "Overall Sketch". A speech bubble from a robot character says "Your sketch looks good to me!". The bottom status bar shows "Updates the feedback and progress meters." and a "select entire glyphs" button.

CogSketch

File Edit View Glyphs Analogy FIRE Experimenter Development Windows Help

4\_Heart-Diagram-Simple / Workspace \*

Layer 1  
human-heart  
Meta-Layer

Right atrium Left atrium  
Blood flows Blood flows  
Right ventricle Left ventricle

Properties Problem Feedback

Your Progress

Update

All Glyphs have Labels

Required Glyphs

Atria

Blood Flow Relations

Ventricles

Overall Sketch

Your sketch looks good to me!

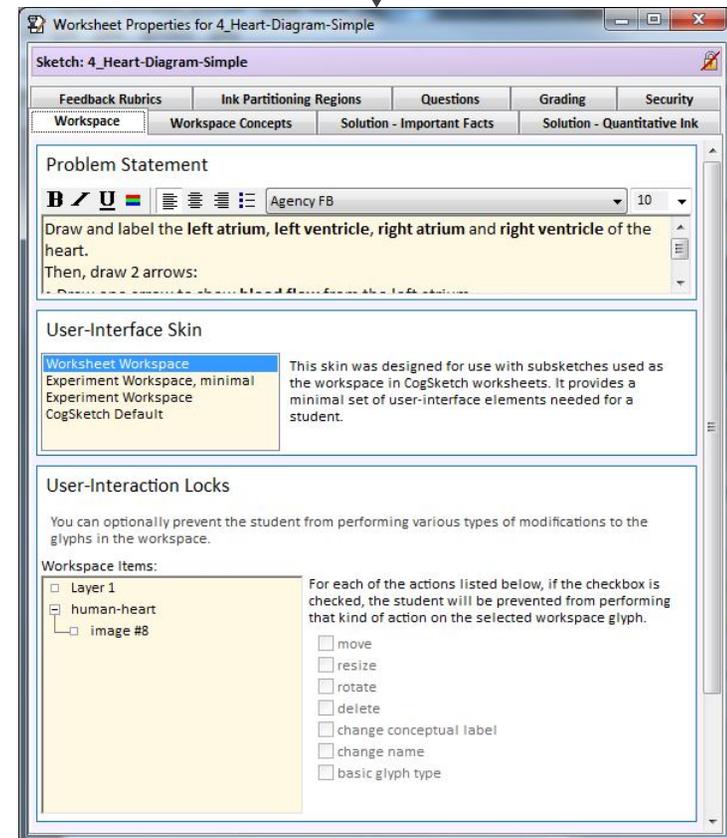
Current Subsketch:  
Workspace  
-14.3, 2.8 in  
zoom = 66%

Updates the feedback and progress meters.

select entire glyphs

# How to Create a Worksheet from Scratch

- Authoring environment for setting up worksheet properties
  - Problem statement, relevant concepts, skin
  - Solution Sketch
  - Correctness Criteria & Feedback
    - Quantitative ink constraints
    - Important qualitative spatial and conceptual facts
  - Grading
  - Password protection



# Quantitative Ink Differences

Rough overlap between student glyph(s) and teacher glyph(s), for when absolute location matters

The image shows a screenshot of the CogSketch software interface. The main window displays a heart diagram with various parts labeled: Right atrium, Left atrium, Right ventricle, and Left ventricle. A blue callout box points to the 'Left atrium' label, containing the text: "Tolerance region defined by the worksheet author, not seen by student".

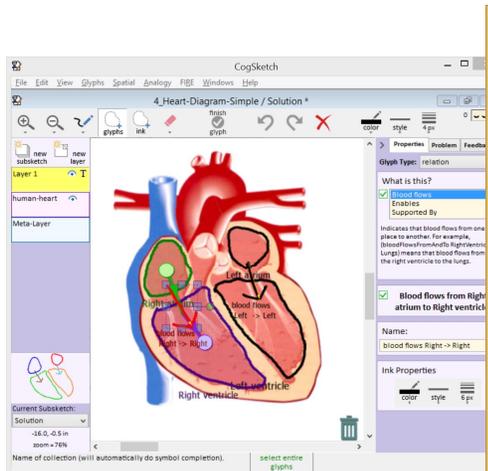
The software interface includes a menu bar (File, Edit, View, Glyphs, Spatial, Analogy, FIRE, Windows, Help) and a toolbar with icons for zooming, drawing, and editing. The left sidebar shows a layer management panel with 'Layer 1' selected. The right sidebar shows a 'Properties' panel for the selected 'Left atrium' glyph, with the following details:

- Properties:** Problem, Feedback
- Glyph Type:** relation
- What is this?:**  Blood flows from Right atrium to Right ventricle. (Indicates that blood flows from one place to another. For example, (bloodFlowsFromAndTo RightVentricle Lungs) means that blood flows from the right ventricle to the lungs.)
- Name:** blood flows Right -> Right
- Ink Properties:** color, style, 6 px

Below the diagram, a 'Workspace Concepts' panel shows a tree structure with 'Left atrium (Object-55)' selected. At the bottom right, a 'Tutoring Advice' box displays the message: "Your drawing of the left atrium isn't quite right." The 'Tolerance' for this comparison is set to 0.75.

# Qualitative Spatial and Conceptual Differences

Sketch facts automatically generated by CogSketch, browsed by worksheet author



The screenshot shows the CogSketch interface with a heart diagram. The diagram is annotated with various facts such as "Blood flows from Right atrium to Right ventricle" and "Left atrium is to the right of Right atrium". The interface includes a menu bar, a toolbar, and a properties panel on the right.

Worksheet Properties for 4\_Heart-Diagram-Simple

Sketch: 4\_Heart-Diagram-Simple

Feedback Rubrics | Ink Partitioning Regions

Workspace | Workspace Concepts | Solution -

Facts Important for Tutoring:

- Blood flows from Right atrium to Right ventricle.
- Blood flows from Left atrium to Left ventricle.
- Left atrium is to the right of Right atrium.

What tutoring advice should be associated with this fact?

The right atrium pumps blood out through a valve. Where does that blood go?

Important facts identified by worksheet author

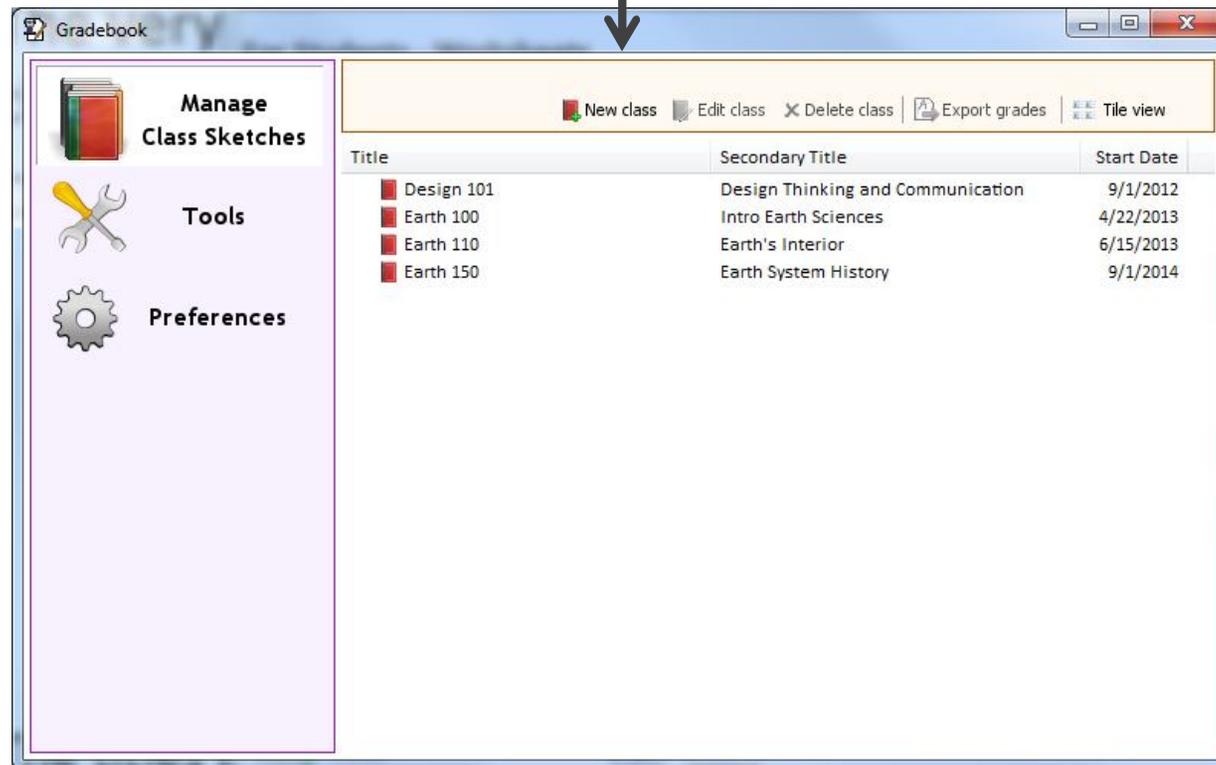
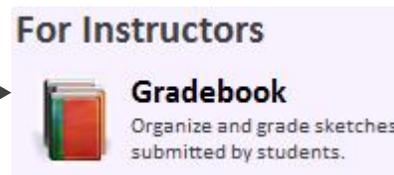
Feedback written by worksheet author

Important facts that are missing or different in student sketch trigger feedback

# Gradebook

- Tool for organizing and grading sketches submitted by students.

From start screen



# Gradebook: grade reports

The screenshot shows a Gradebook application window. On the left, there are sections for 'Manage Class Sketches' and 'Tools'. The main area displays a table with columns for Title, Student(s), Date Received, and Score. A yellow box highlights the 'View grade report' button in the top right. An arrow points from this button to a 'Total Score' section at the bottom left, which shows a normalized score of 44.4 / 100 points and a raw score of 16 / 36 points.

Title	Student(s)	Date Received	Score
earth orbit example	<solution sketch>	8/21/2014	
earth orbit example -- gw	George Washington	8/21/2014	44.4 / 100
	la	8/22/2014	100.0 / 100
	th	8/22/2014	100.0 / 100
	an	8/22/2014	16.7 / 100

**Total Score**

normalized: 44.4 / 100 points

raw: 16 / 36 points

## Scoring Details

### Missing Glyphs

For each of the glyphs listed below, points are awarded if the student has included the glyph in their sketch.

Student Score: 10 / 10 points

- [5 points] Orbit
- [5 points] Sun

Scoring details, including what advice would be given to the student's final sketch

### Non-Quantitative Facts Important for Tutoring

The following are the facts marked as important for tutoring that don't mention quantitative values. Points are awarded if the tutor doesn't find anything wrong with the corresponding facts in the student's sketch.

Student Score: 0 / 20 points

- [0 points]

Correct Answer would be:

(objectContains "Orbit" "Sun")

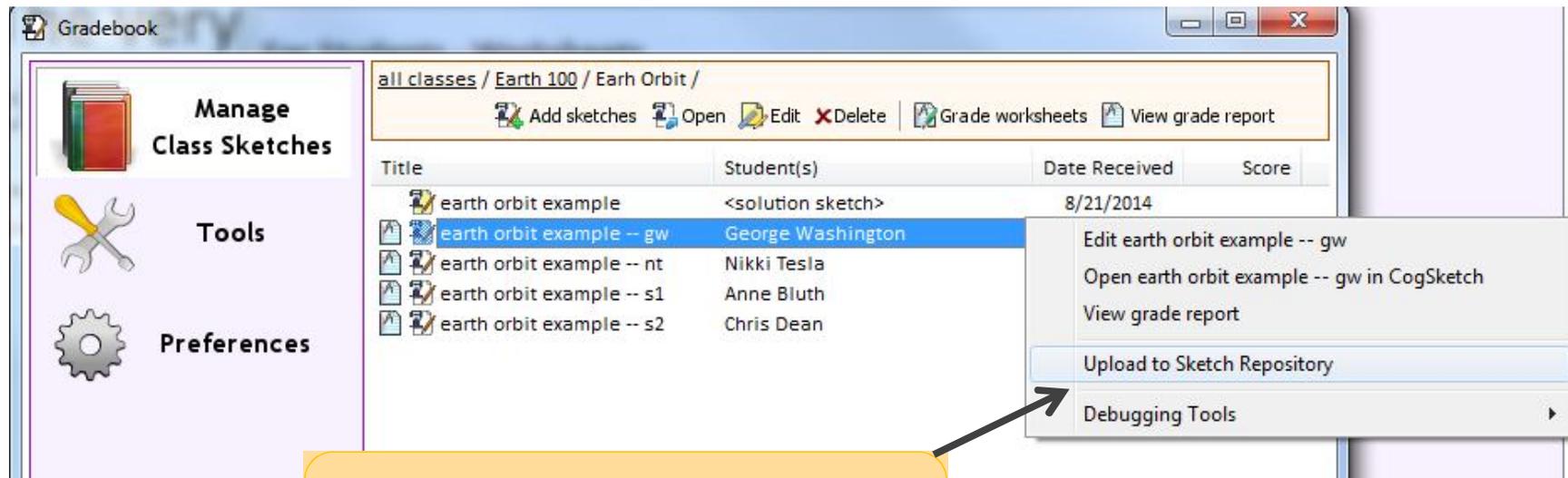
Student had the following similar facts:

(objectContains "orbit" "my home")

Tutor Says: Shouldn't the Earth orbit around the Sun?

# Sending Sketches to the CogSketch Team

- Sketches **anonymized on-site** to protect privacy
- Anonymized sketches can be used for
  - Cognitive science experiments/analyses
  - Helping us to improve the software and user experience

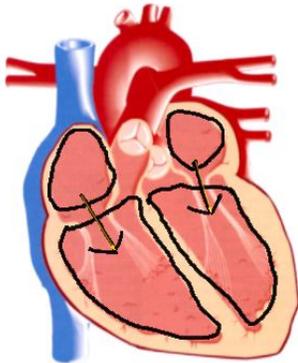


Right-click on a sketch to anonymously send it back to us

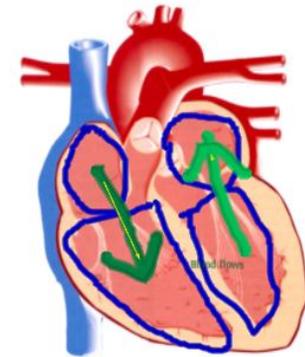
# Sketch Worksheets: domain-general, model-based tutoring

(Yin et al. 2010)

Teacher's Solution



Student's Solution



Structure-Mapping Engine

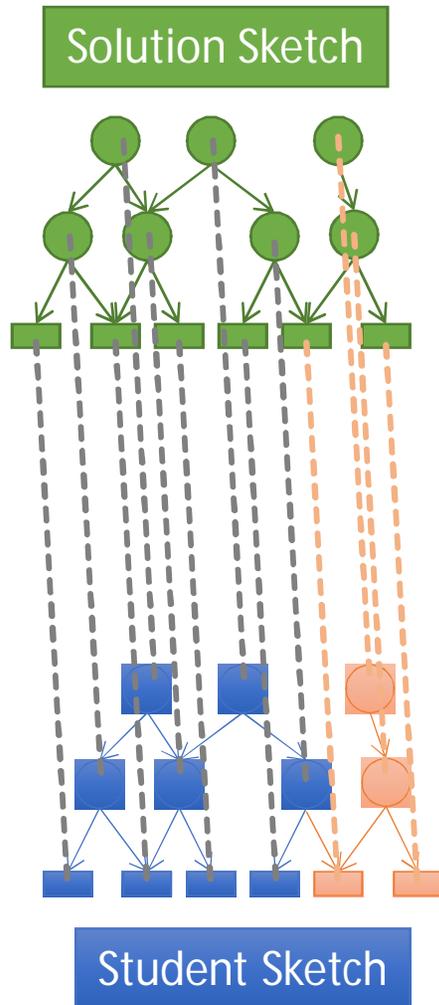
Differences between sketches +  
Pre-determined correctness criteria

Feedback to Student:  
*"The left atrium pumps blood out through a valve.  
Where does that blood go?"*

*How does the comparison, feedback generation work?*

# Structure Mapping Engine (SME)

(Falkenhainer et al. 1989; Forbus et al. 1994)

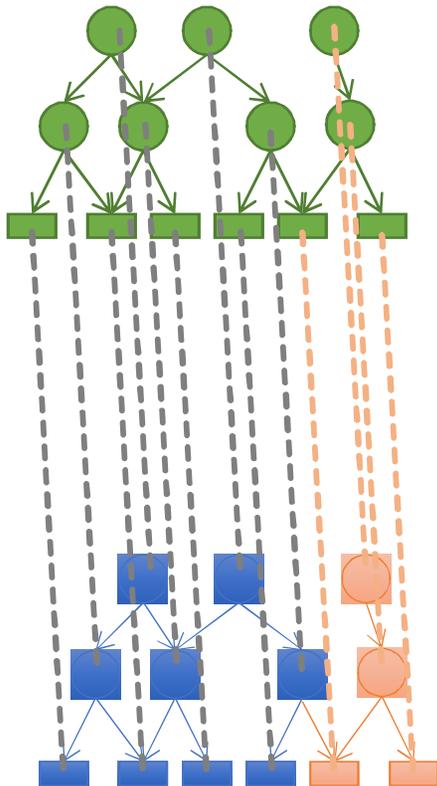


- Takes two structured descriptions (base, target)
- Produces one or more mappings:
  - • Correspondences, e.g. student's right atrium corresponds to teacher's right atrium
  - • Candidate inferences, e.g. an important fact in the solution is missing from the student's sketch
- Enables feedback for
  - Quantitative ink differences
  - Qualitative spatial or conceptual differences

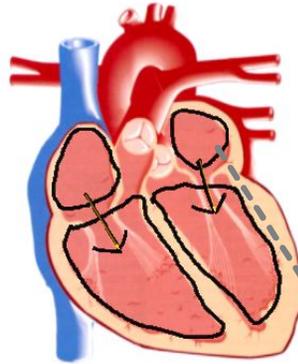
# Structure Mapping Engine (SME)

(Falkenhainer et al. 1989; Forbus et al. 1994)

Solution Sketch

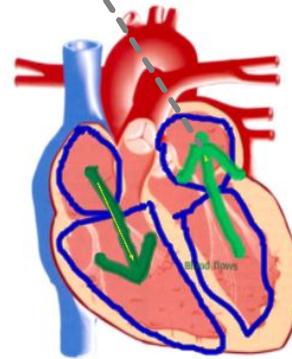


Student Sketch



For a given student glyph, does absolute location matter?

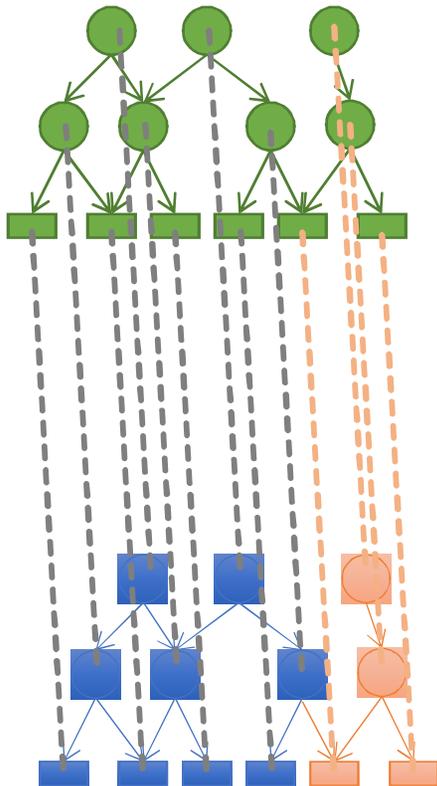
SME correspondence indicates the corresponding solution glyph and the corresponding tolerance region



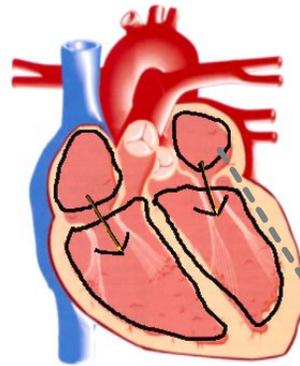
# Structure Mapping Engine (SME)

(Falkenhainer et al. 1989; Forbus et al. 1994)

Solution Sketch



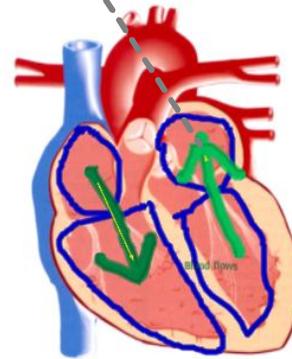
Student Sketch



Are there any important qualitative differences?

**Important fact:**  
Blood flows from *left atrium* to *left ventricle*

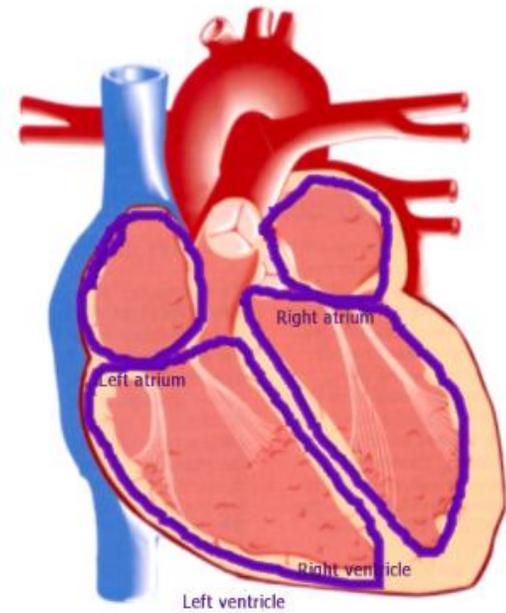
SME candidate inference indicates an important difference



**Important fact:**  
Blood flows from *left ventricle* to *left atrium*

# Match Constraints

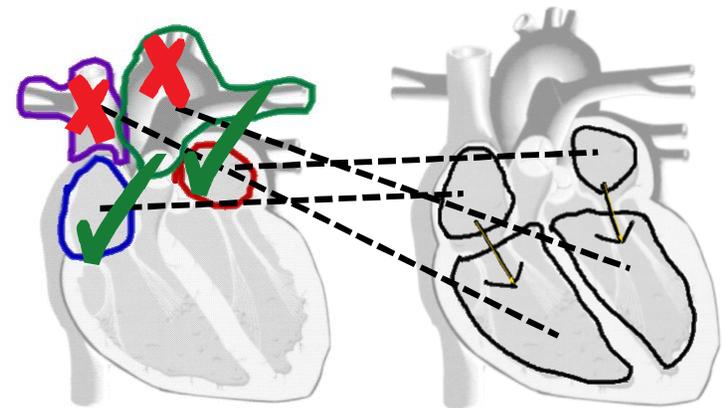
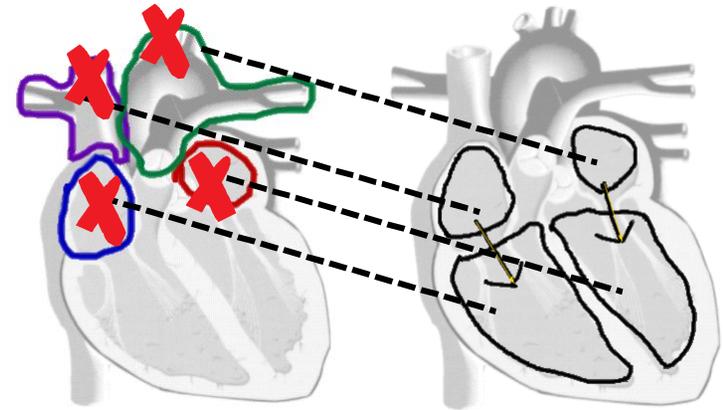
- *Partition constraints*
  - If two entities correspond to each other, they must have identical attributes
  - Prevents matches suggested by spatial arrangement, but incorrect because of labels
- *Quantitative ink constraints*
  - When multiple matches are possible, favor ones that satisfy quantitative ink constraints



Visually, this looks correct, but what if the student has left and right mixed up?  
**Labels matter!**

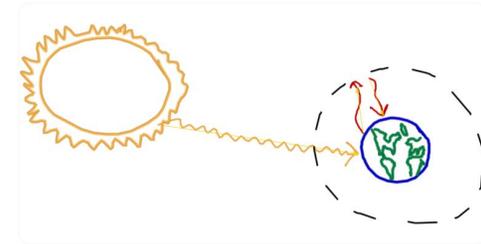
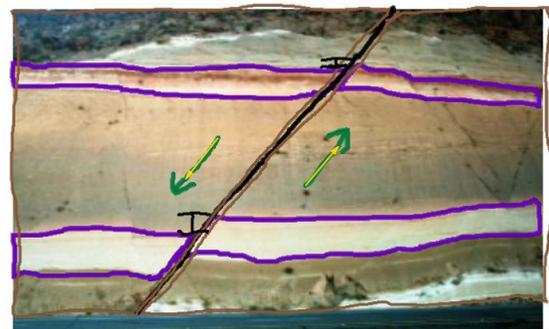
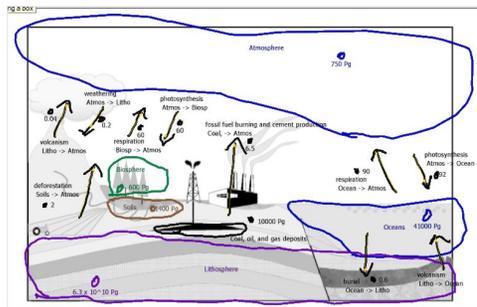
# Match Constraints

- *Partition constraints*
  - If two entities correspond to each other, they must have identical attributes
  - Prevents matches suggested by spatial arrangement, but incorrect because of labels
- *Quantitative ink constraints*
  - When multiple matches are possible, favor ones that satisfy quantitative ink constraints



# Worksheet Pilot Studies

- Objectives
  - Gather data needed to improve representations, algorithms, and user experience
  - Understand how to make worksheets practical in classrooms and for homework assignments



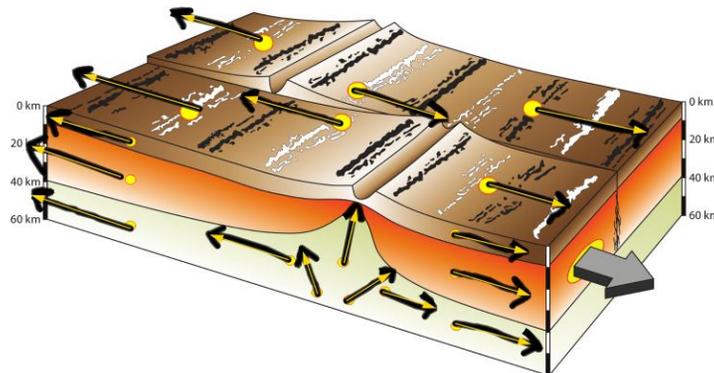
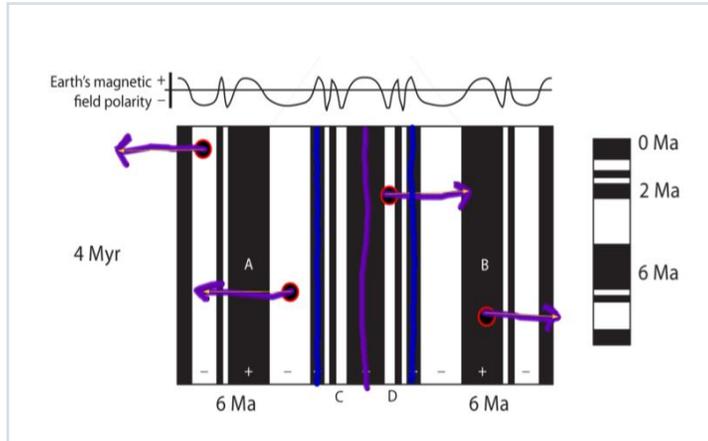
Northwestern University, Earth 201,  
Profs. Brad Sageman, Fall 2009

- Extra credit assignment: fault identification worksheets (3); 10 students
- Mandatory homework assignment: carbon cycle worksheet; 28 students

Northwestern University, Earth 201, Prof.  
Andrew Jacobson, Spring 2011

- Mandatory homework assignment: fault identification worksheets (3); 40 students
- Extra credit assignment: carbon cycle and greenhouse worksheets; 27 students

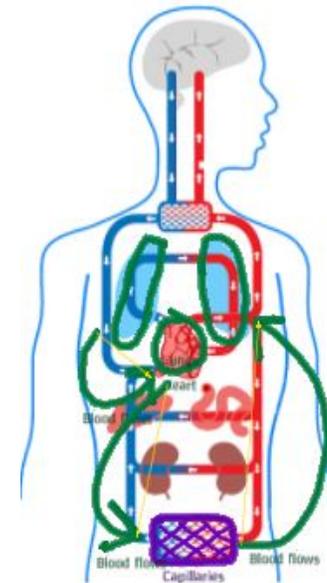
# Worksheet Pilot Studies (cont'd)



- Spring 2011, Carleton College, Geo 110
  - Four plate tectonics worksheets as part of in-class group activity
  - 1 hour lab section
  - 2-4 students per laptop
  - Students completed CogSketch tutorial and completed on average 2 worksheets per group in < 1 hour
  - Students had never seen/used CogSketch before

# Worksheets in the Classroom – Evidence of Learning Gains

- Westampton Elementary School 5<sup>th</sup> graders, n = 50
- Four worksheets on circulatory system function, after lesson
  - Self-paced, 1 hour time limit
  - 70% of students completed all four
- Three measures of circulatory system understanding
  - Heart chamber identification (fill in the blanks on diagram)
  - Blood flow order (list order on diagram)
  - Flow of oxygen (multiple choice questions)



Significant learning gains on 2 out of 3 measures

\*Collaborators: Brian Miller (Towson University), Jennifer Cromley (Temple University)

# Worksheets in the Classroom – What did the 5<sup>th</sup> graders think?

*CogSketch survey Likert-style scale-item responses*

Question	<i>M</i>	<i>SD</i>
CogSketch was easy to learn by practicing with the person, cat, and dog.	5.06	1.42
I liked using CogSketch.	5.87	1.71
CogSketch is easy to use.	5.13	1.52
I would feel comfortable using a CogSketch for a class assignment.	5.65	1.89
CogSketch helped me learn about the circulatory system.	5.60	1.75

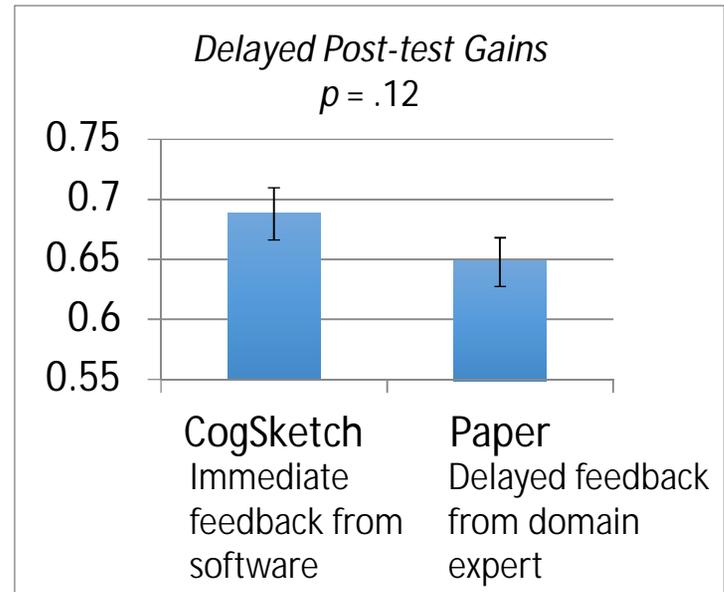
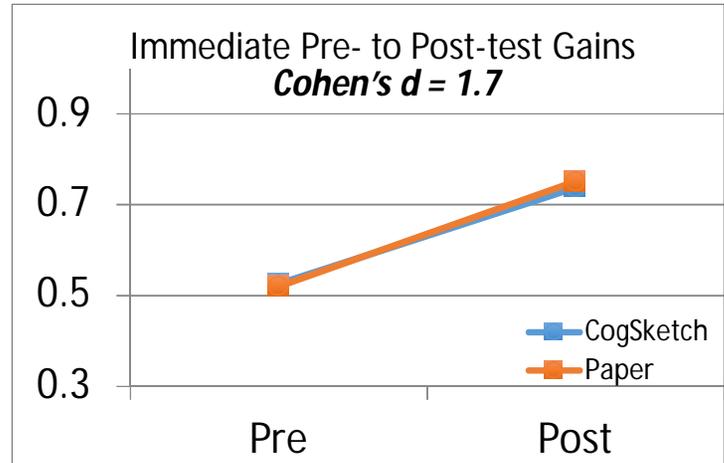
Note: all questions have a maximum score of 7

# Worksheets in the Classroom – Undergraduate Geoscience at UW Madison

- Worksheets developed by domain expert: Bridget Garnier (geoscientist and instructor, *not a computer scientist*)
- Two groups
  - CogSketch group (n = 64) used CogSketch worksheets, received immediate, on-demand feedback from tutor
  - Paper group (n = 93) did identical exercises on paper, received detailed feedback from Bridget Garnier 1 week later
- Note: in real classrooms, students **do not** receive paper worksheets because they are too time consuming to grade!
- Pre-test prior to completing worksheet
- Immediate post-test after completing worksheet
- Delayed post-test (covering all 16 topics) at end of semester

# Worksheets in the Classroom – Undergraduate Geoscience Results

- All students made large, significant learning gains from pre- to immediate post-test
  - 12/16 worksheets: no difference
  - 3/16 worksheets: CogSketch better than paper
  - 1/16 worksheets: paper better than CogSketch
- Further analyses needed to understand what makes some worksheets more/less effective than others
- Overall, **no significant difference** between CogSketch and paper on delayed-post tests
- Paper condition  $\neq$  business as usual!

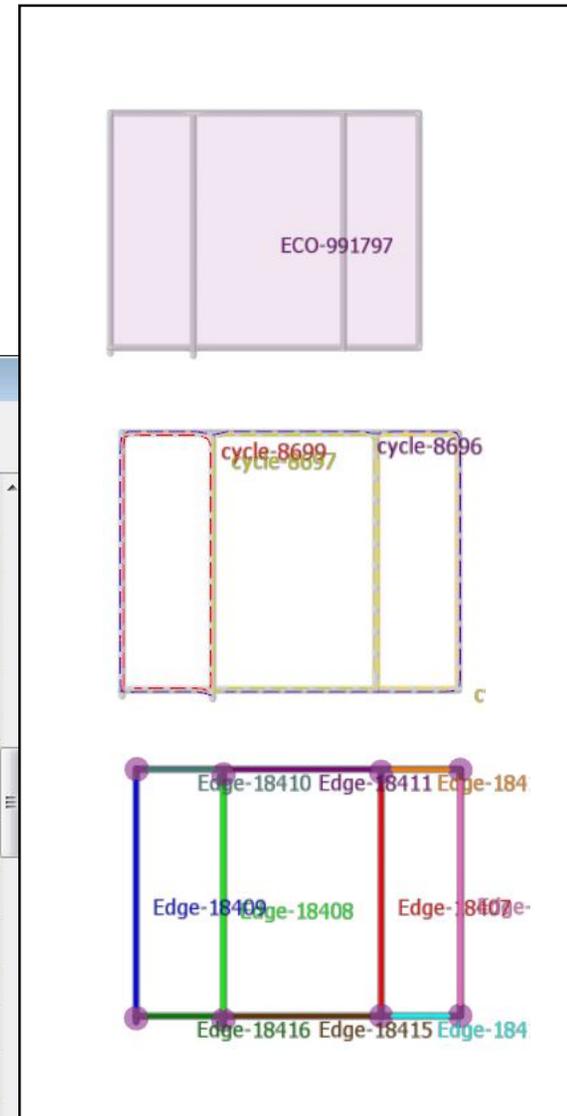
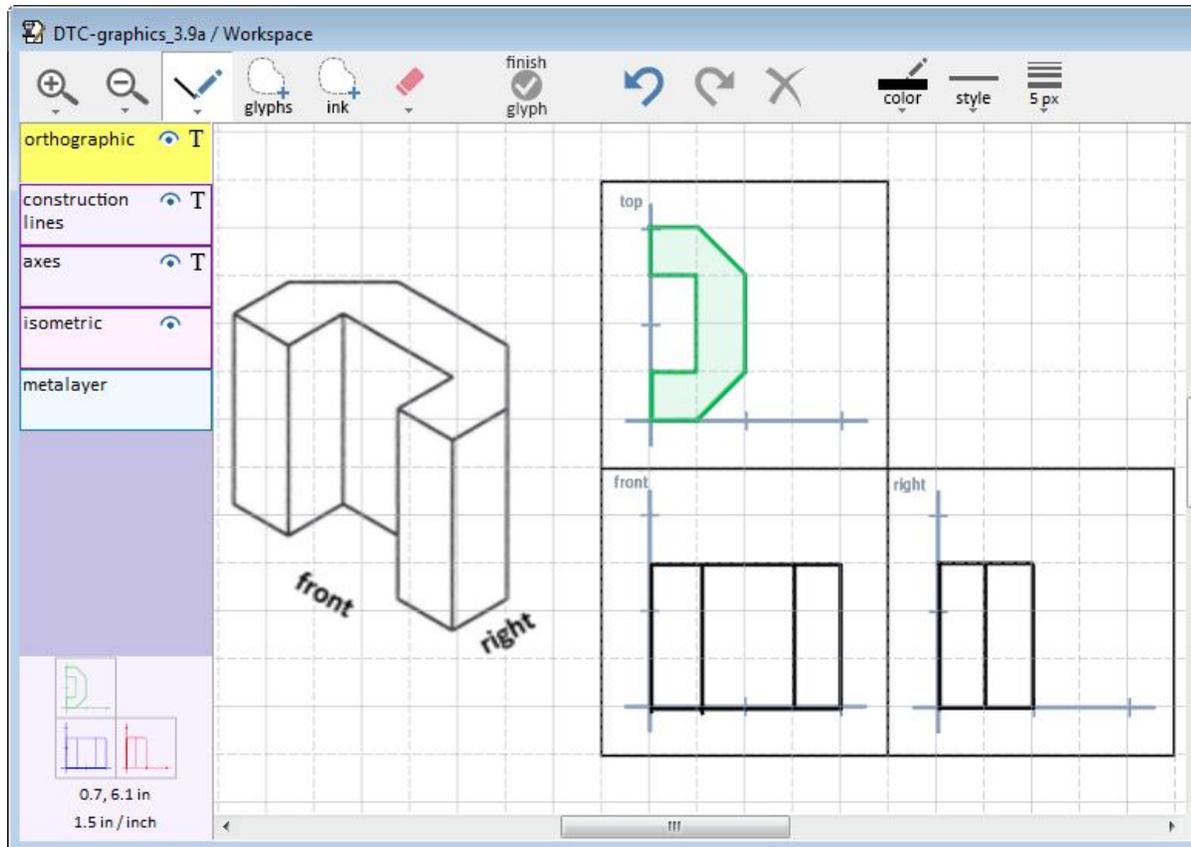


CogSketch potential:

*Enabling instructors to give effective sketching assignments that are too impractical to implement on paper.*

# Design Thinking and Communication

## Orthographic Projection Sketches



# Worksheets: Technical Lessons

- Two types of spatial representations go a long way
  - Qualitative positional and topological relations
  - Quantitative ink constraints
- But, for things like orthographic projection, more detailed representations needed
- Tight constraints on analogical mapping are necessary, since the variance of student drawings is huge
- Lower entry-barriers to authoring key to scaling up

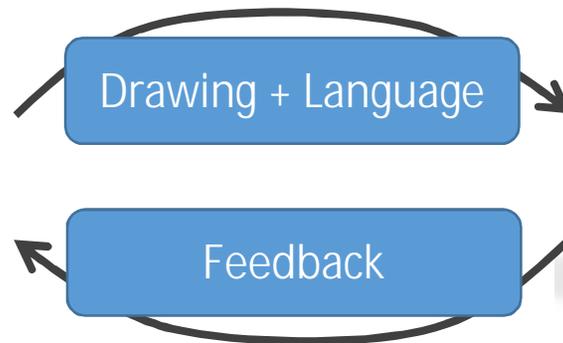
# Design Coach: Setting and Problem

Design Thinking and communication course at Northwestern University



**Problem: students have trouble using sketches to communicate**

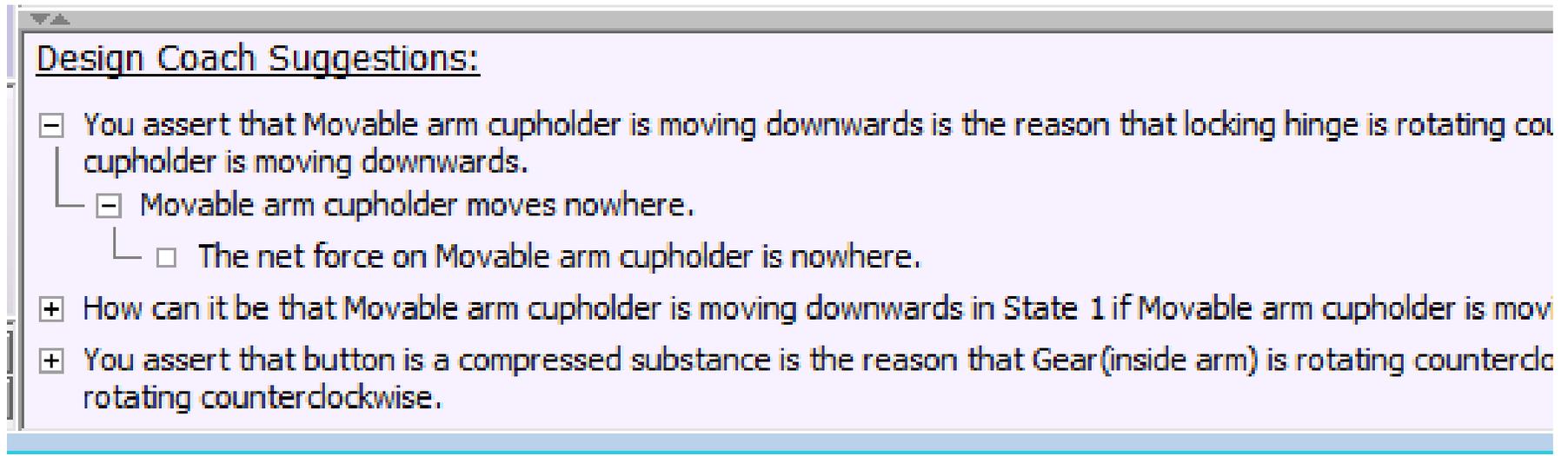
# Design Coach: domain-specific, first principles-based tutoring



- For students to practice design explanation with sketching and language
  - Student supplements sketch with language-like input using restricted syntax templates
  - Qualitative reasoning allows coach to understand physical mechanisms, i.e. support forces, motion, springs, gears, pulleys, cords
  - Coach provides feedback when explanation is unclear
- Domain-specific (mechanics and design) knowledge enable more in-depth tutoring

# Giving Students Feedback

- Design Coach checks for
  1. Unexplained or impossible motions depicted in the sketch [Wetzel & Forbus, 2008]
  2. Unsupported or contradictory template-based sentences [Wetzel & Forbus, 2009]
  3. User input errors

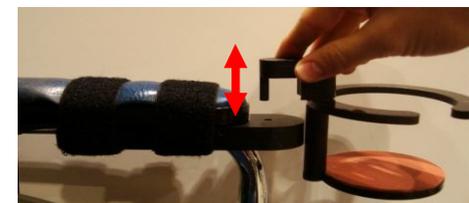
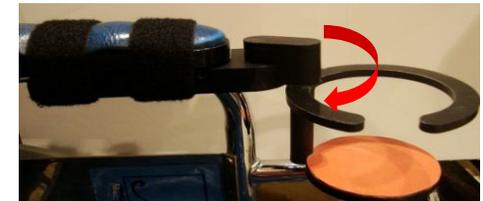


The image shows a screenshot of a software window titled "Design Coach Suggestions:". The window has a light purple background and a blue border. It contains a list of suggestions with a hierarchical structure indicated by square icons and lines. The suggestions are:

- You assert that Movable arm cupholder is moving downwards is the reason that locking hinge is rotating clockwise and Movable arm cupholder is moving downwards.
  - Movable arm cupholder moves nowhere.
    - The net force on Movable arm cupholder is nowhere.
- How can it be that Movable arm cupholder is moving downwards in State 1 if Movable arm cupholder is moving upwards?
- You assert that button is a compressed substance is the reason that Gear (inside arm) is rotating counterclockwise and Gear (inside arm) is rotating counterclockwise.

# Design Coach in the Lab and Classroom

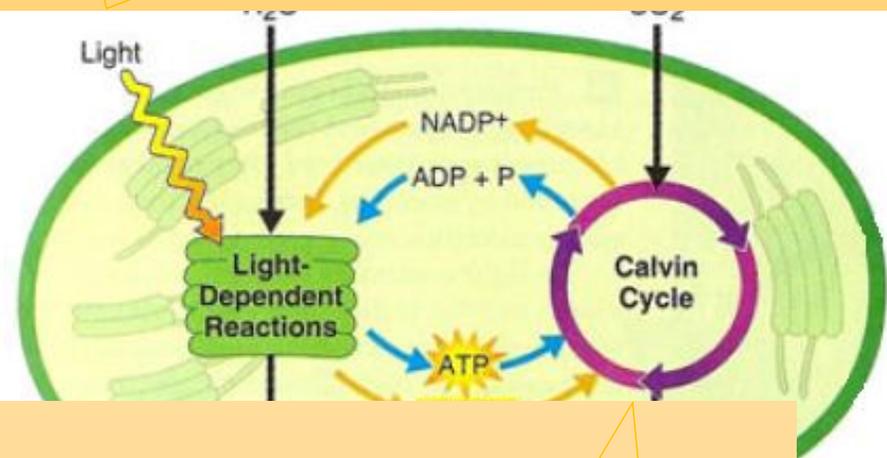
- Design Coach activity
  - Design Thinking and Communication (DTC) students: Fall 2012, Winter 2013, and Fall 2013
  - Sketch and explain how a given device works
  - Devise a refinement and explain it to Design Coach
- Sketching Anxiety Survey
  - Based on math anxiety survey by Beilock et al. (2010)
  - Given pre- and post-activity
- Design Coach significantly reduced sketching anxiety in two out of three quarters



# Sketching as an Assessment Tool (without immediate feedback)

- Simple tasks, e.g. copying, tracing can distinguish experts and novices
- Pilot study from Louis Gomez (UCLA) on copying biology process diagrams
  - 10 non-science majors
  - 10 pre-med students

Experts start at beginning of process

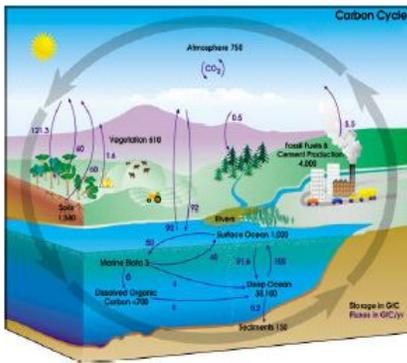


Novices start with visually salient features

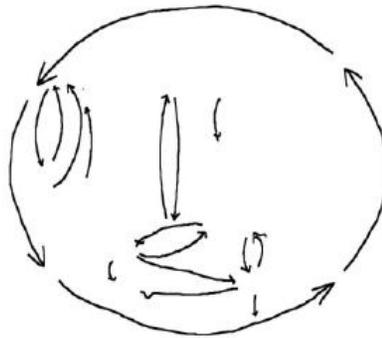
# Sketching as an Assessment Tool (without immediate feedback)

Jee et al. (2009): Geoscience undergrad/grad students include more relations compared to psychology undergrads

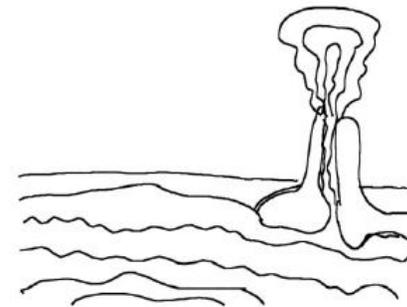
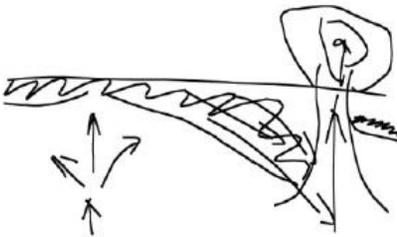
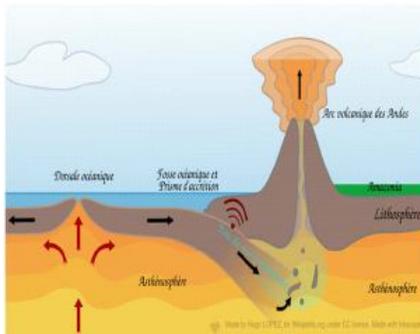
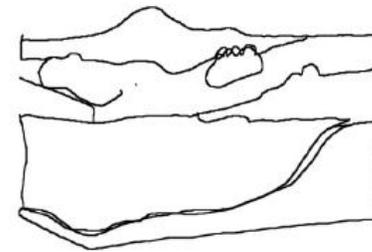
Geo Diagram



Geo Student sketch

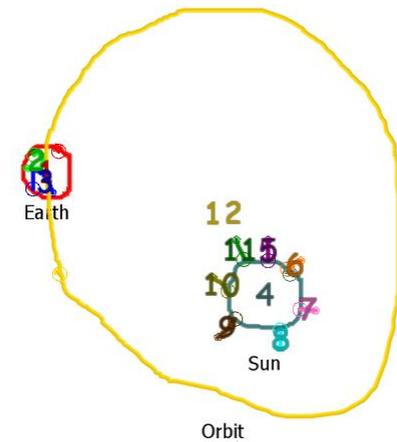
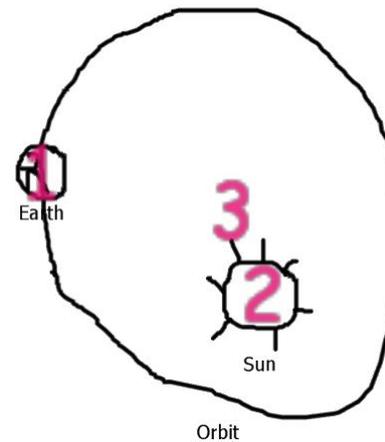


Novice sketch

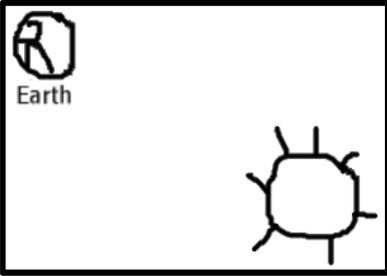


# Other Analysis Tools

- Timing data
  - View glyph ordering
  - View ink stroke ordering
  - Sketch playback
- History data
  - Detailed history and ink data exported to comma separated values (\*.csv) format
  - HTML reports detailing user actions with screenshots



- 48) Created glyph #7820 ("Sun")
  - Start:
    - clock time: 2014-08-22 15:43:03.174
    - sketch timestamp (seconds): 2178.186
  - Finish:
    - clock time: 2014-08-22 15:43:19.092
    - sketch timestamp (seconds): 2194.104
  - Elapsed time (seconds): 15.918
  - user / source: George Washington



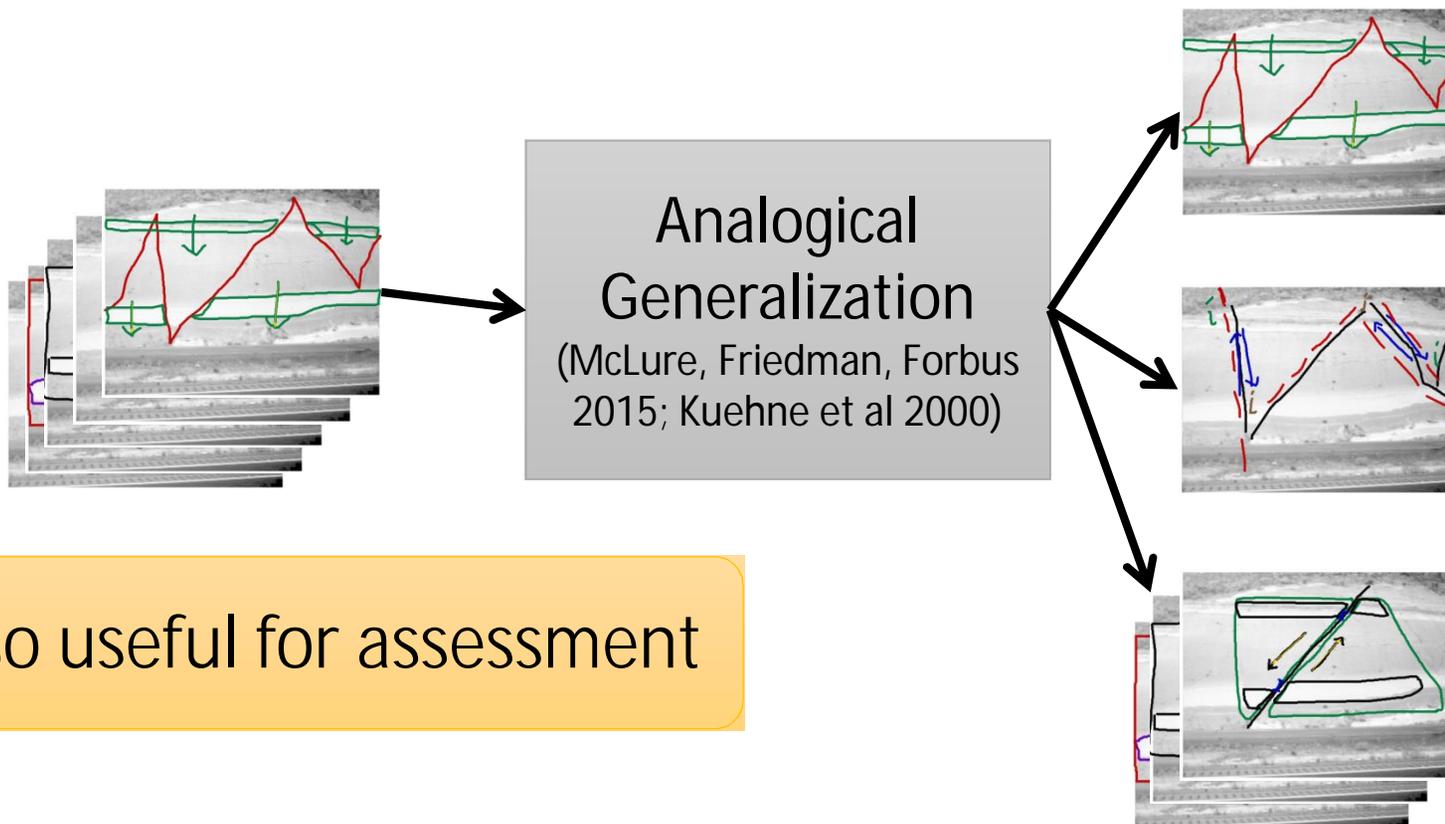
- 49) Changed conceptual labels for glyph #7820 ("Sun")
  - clock time: 2014-08-22 15:43:20.729
  - sketch timestamp (seconds): 2195.741
  - user / source: George Washington
  - Removed: Entity
  - Added: Sun
  - All conceptual labels assigned at this time: Sun

Useful assessment data

# Analysis tools for the future

## Sketch clustering for common answer patterns

(see Chang & Forbus 2014 *AI Magazine* for details)



Also useful for assessment

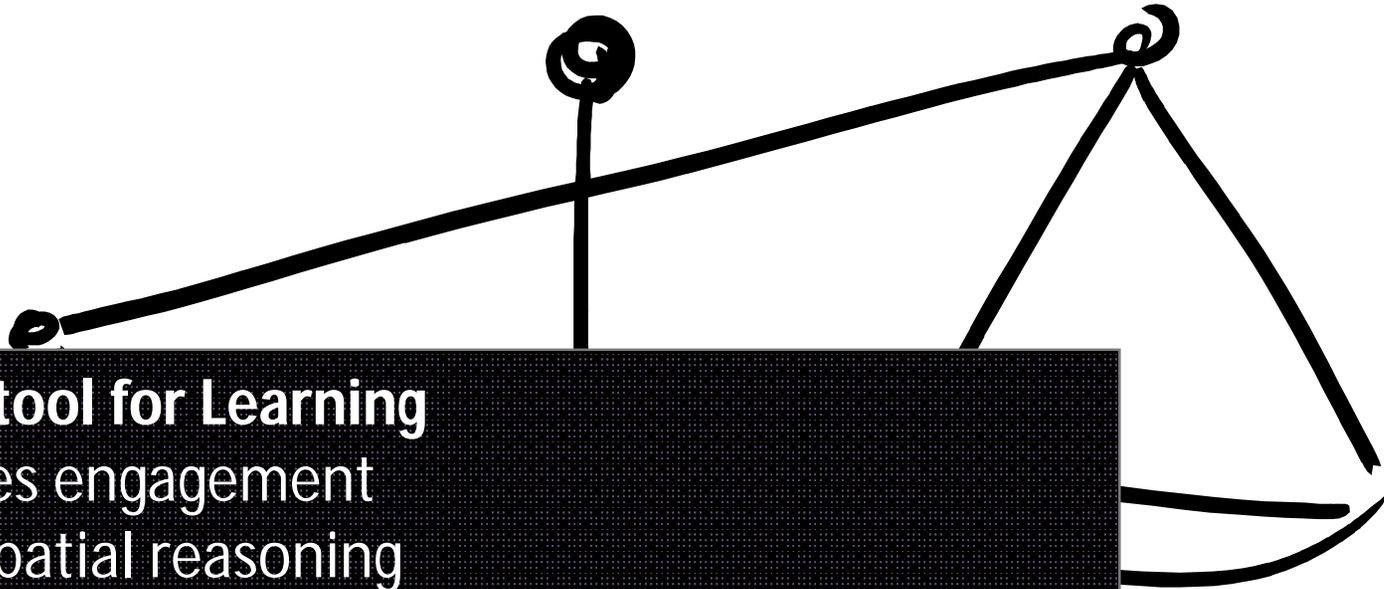
# CogSketch and Education: Future Plans

- Further analyses of classroom data to determine best practices for developing worksheets and opportunities for improved tutoring
- Worksheet web exchange: online community for instructors to share worksheets
- Cloud-based version of CogSketch, potential for use on lightweight machines (e.g. Android devices, iPad)
- Continue to collaborate with domain experts and educators

# Summary

- Worksheets are designed to help students learn spatial phenomena, especially layouts
  - In 5<sup>th</sup> grade science data, evidence of learning beyond that of regular instruction
  - In undergraduate geoscience data, just as good as paper with detailed delayed feedback
- Design Coach uses qualitative physical reasoning to help students with design explanations
  - Evidence that CogSketch can be combined with other AI techniques to provide sophisticated tutoring

# Tip the scale!



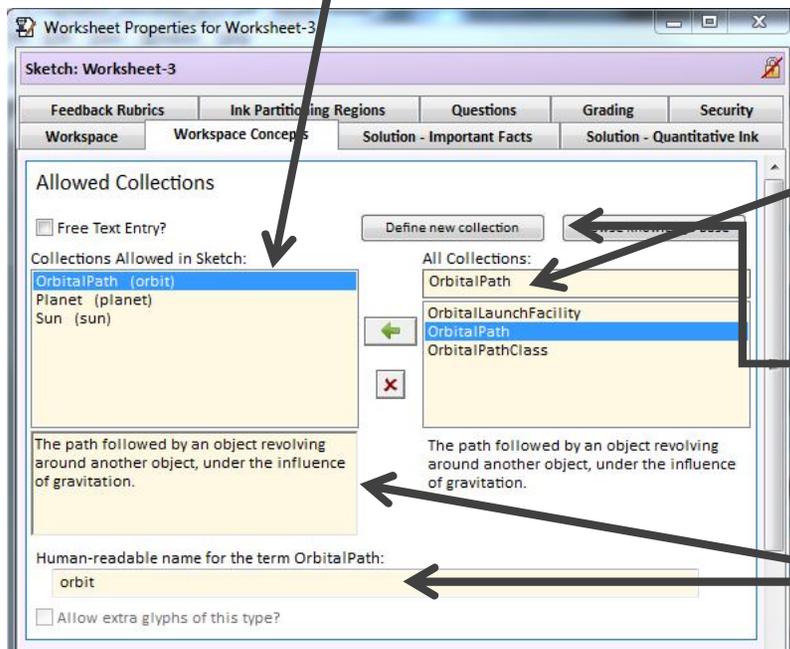
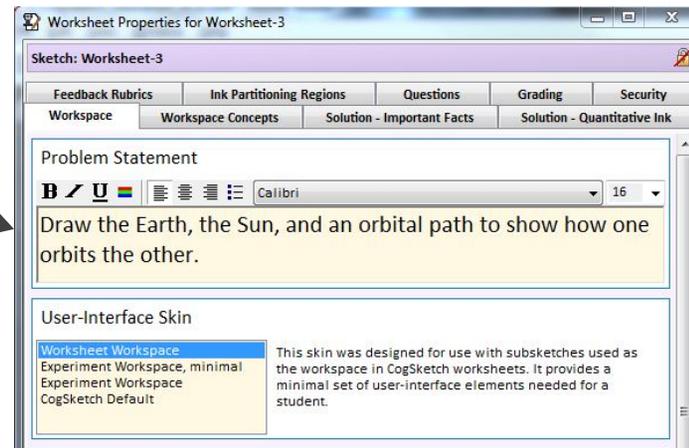
## **Powerful tool for Learning**

- Increases engagement
- Easier spatial reasoning
- Multi-modal reasoning
- **On-demand feedback from CogSketch**
- **Faster grading via CogSketch**
- **More sketching exercises for students learning about science**

# Backup Slides

# 1. Set up the Workspace

- Problem statement, i.e. instructions for the student
- Workspace concepts, i.e. labels for student glyphs



Search the knowledge base for concepts and select the ones you want

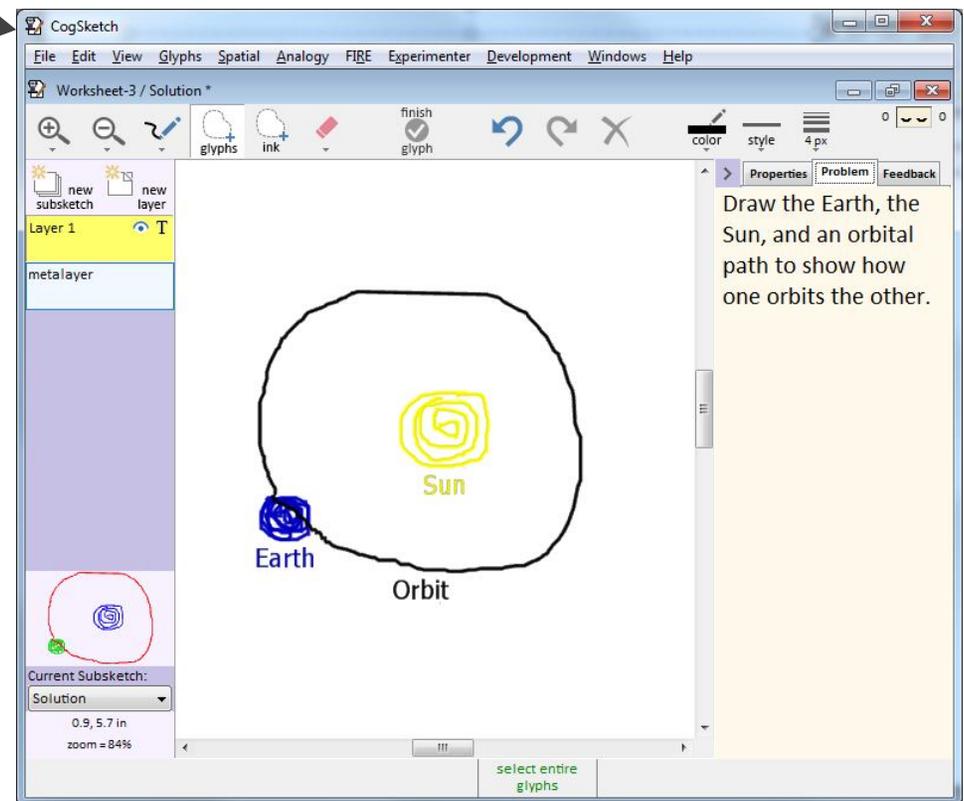
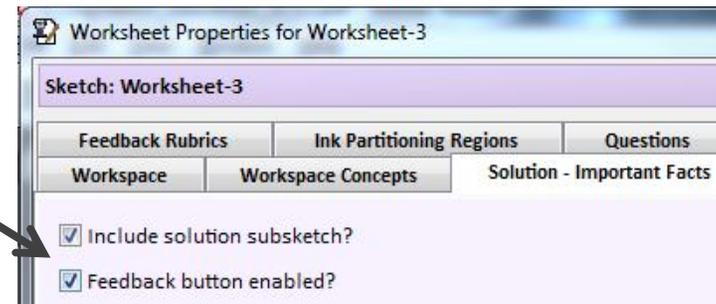
Don't see the concept you want? Define a new one for this worksheet.

Define new collection

Edit name strings and descriptions, control over what the student sees

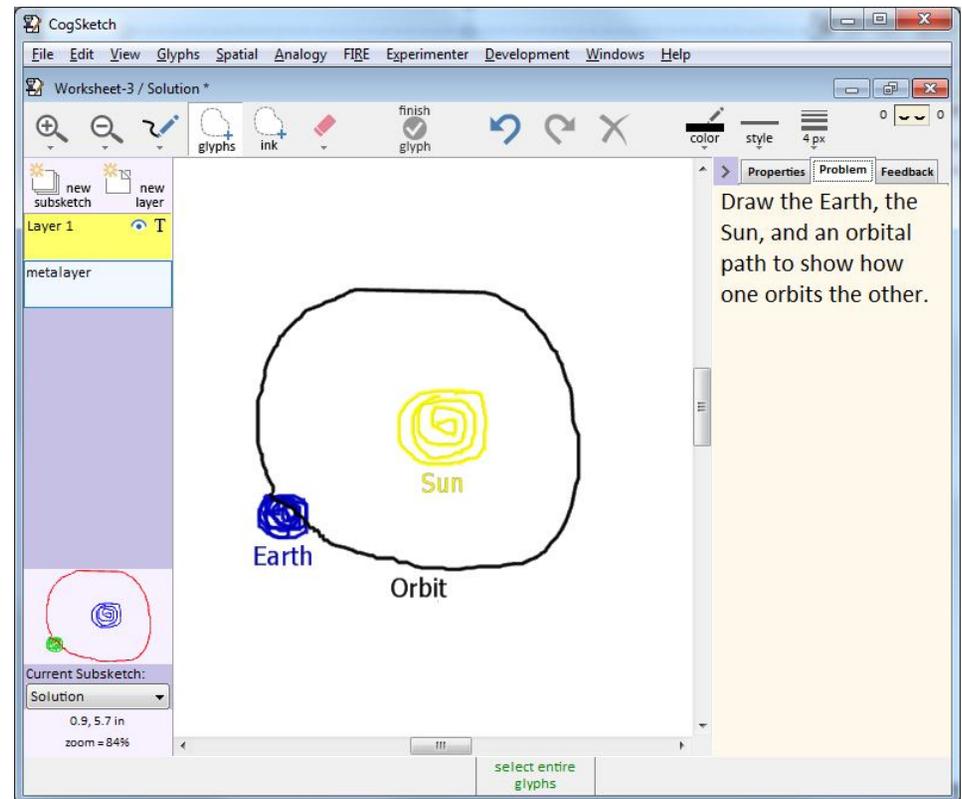
## 2. Draw Worksheet Solution

- Include solution sketch and feedback
- Draw and label solution into CogSketch
  - Draw the sun, label it *Sun*
  - Draw the Earth, label it *Planet*
  - Draw the Orbit, label it *Orbit*
  - Optional: change name strings, e.g. Planet → Earth



## 2. Draw Worksheet Solution

<Representations  
generated by cogsketch  
automatically>



### 3. Correctness Criteria & Feedback

Facts about the solution are the representations generated by CogSketch

Worksheet Properties for Worksheet-3

Sketch: Worksheet-3

Workspace Concepts    Solution - Important Facts    Solution - Quantitative Ink

Ink Partitioning Regions    Questions    Grading    Security

...n subsketch?  
...n enabled?

...ce: Facts Important for Tutoring

Selecting a glyph in the "Solution Glyphs" section below will show facts about the glyph that might be useful for tutoring. Choose which facts you want to be considered important.

Solution Items:    Facts About Selected Item    Get more facts...

- Layer 1
  - Sun (Object-7766)
  - Earth (Object-7767)
  - Orbit (Object-7768)

Facts About Selected Item:

- Glyph of Sun is inside of the glyph-group with glyph of Orbit (the container) and and glyph of Sun (the thing(s) inside).
- Sun is a sun.
- Glyph-group with glyph of Orbit (the container) and and glyph of Sun (the thing(s) inside) is a ContainedGlyphGroup.
- Glyph of Sun is contained by glyph of Orbit.**
- Sun is to the right of Earth.
- Glyph of Sun is to the right of glyph of Earth.

Facts Important for Tutoring:

- Glyph of Sun is contained by glyph of Orbit.**

Select facts that are important for getting the worksheet correct

### 3. Correctness Criteria & Feedback

For each important fact, write a piece of advice for it.

*If the student's sketch is missing the important fact, they will get the advice.*

If the student's orbit glyph does *not* contain the Sun glyph, then the tutor will say, "Shouldn't the Earth orbit around the Sun?"

The screenshot shows a window titled "Worksheet Properties for Worksheet-3". Below the title bar, there is a section titled "Facts Important for Tutoring:" with a green downward arrow. A list of facts is displayed, with the first item "Glyph of Sun is contained by glyph of Orbit." highlighted in blue. Below the list is a text input field with the prompt "What tutoring advice should be associated with this fact?". The text "Shouldn't the Earth orbit around the Sun?" is entered into this field. A yellow callout box on the left points to the highlighted fact and the text input field.

# 4. Grading Rubrics

Multiple grading criteria available (scroll down for more)

The screenshot shows a software interface for creating a grading rubric. At the top, there are tabs for 'Concepts', 'Solution - Important Facts', 'Solution - Quantitative Ink', 'Feedback Rubrics', 'Ink Partitioning Regions', 'Questions', 'Grading', and 'Security'. The 'Grading' tab is active. Below the tabs, there is a 'Normalize To:' field set to '100' points, with '(36 points total before normalization)' in parentheses. The main content area is divided into sections. The first section is 'Missing Glyphs' with a total of '10 points [27.8%]'. It contains a description and an 'Update' button. Below this is a table with two rows: 'Orbit (Object-7773)' with a point value of 5, and 'Sun (Object-7771)' with a point value of 5. The second section is 'Non-Quantitative Facts Important for Tutoring' with a total of '20 points [55.6%]'. It contains a description and an 'Update' button. Below this is a table with one row: 'Glyph of Sun is contained by glyph of Orbit.' with a point value of 20.

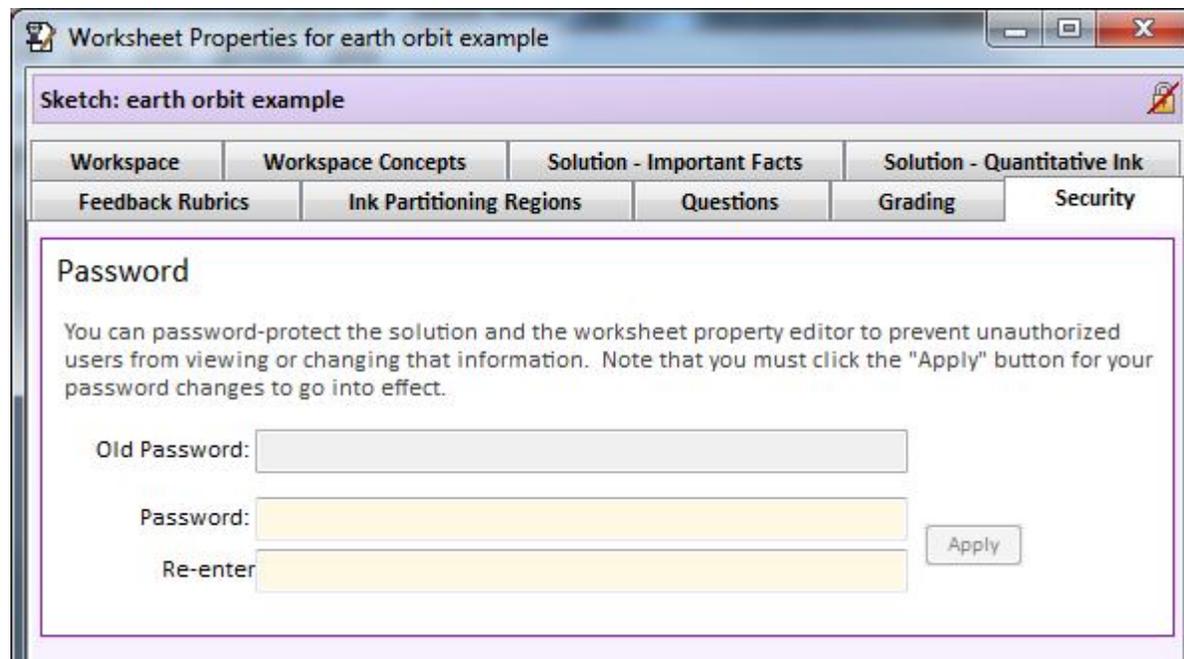
Points are normalized automatically

List of important items and facts are listed automatically

You assign point values for each item

## 5. Password

- Prevents students and/or research participants from viewing the solution



Worksheet Properties for earth orbit example

Sketch: earth orbit example

Workspace	Workspace Concepts	Solution - Important Facts	Solution - Quantitative Ink
Feedback Rubrics	Ink Partitioning Regions	Questions	Grading
			Security

### Password

You can password-protect the solution and the worksheet property editor to prevent unauthorized users from viewing or changing that information. Note that you must click the "Apply" button for your password changes to go into effect.

Old Password:

Password:

Re-enter:

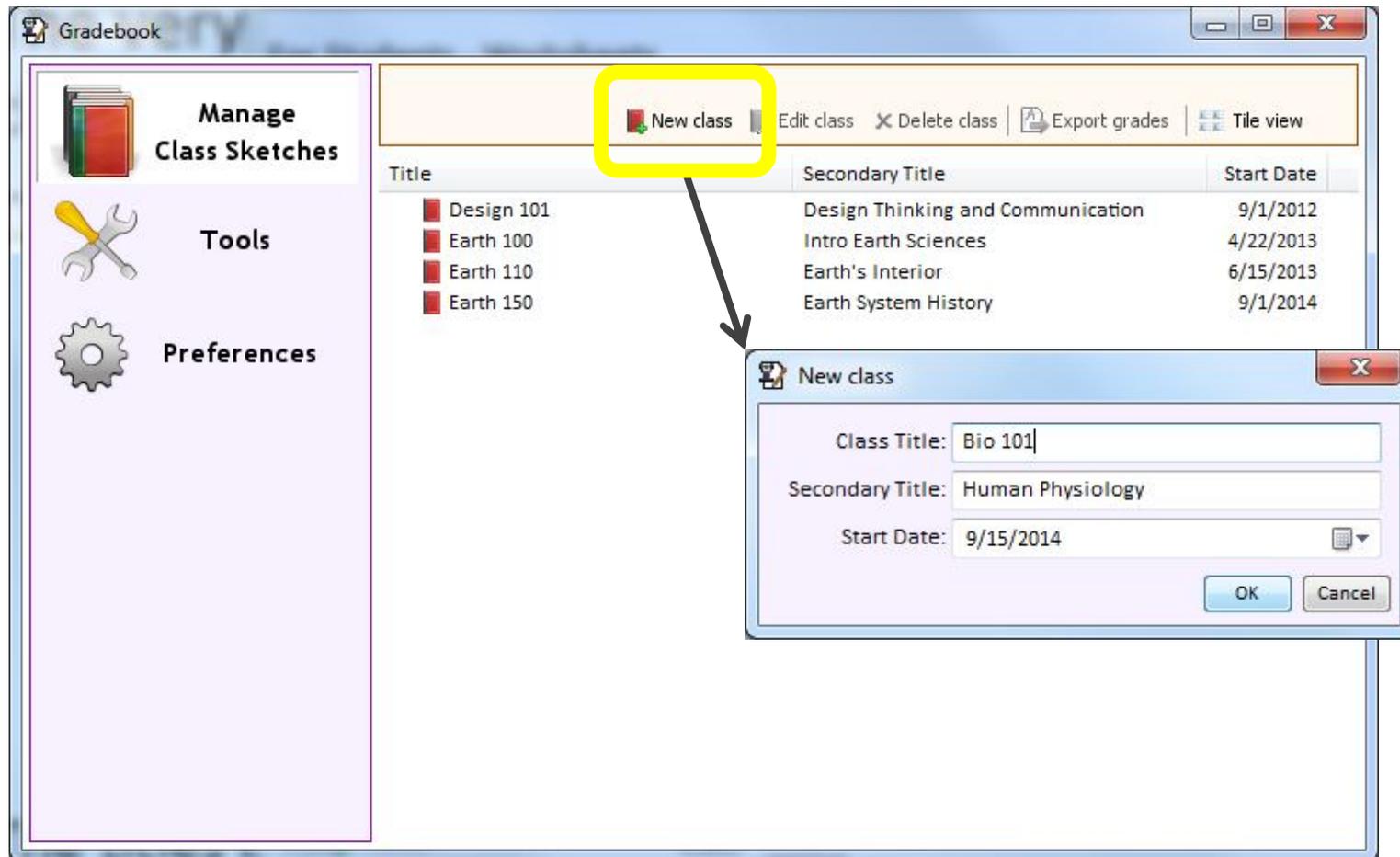
Apply

## 6. Testing

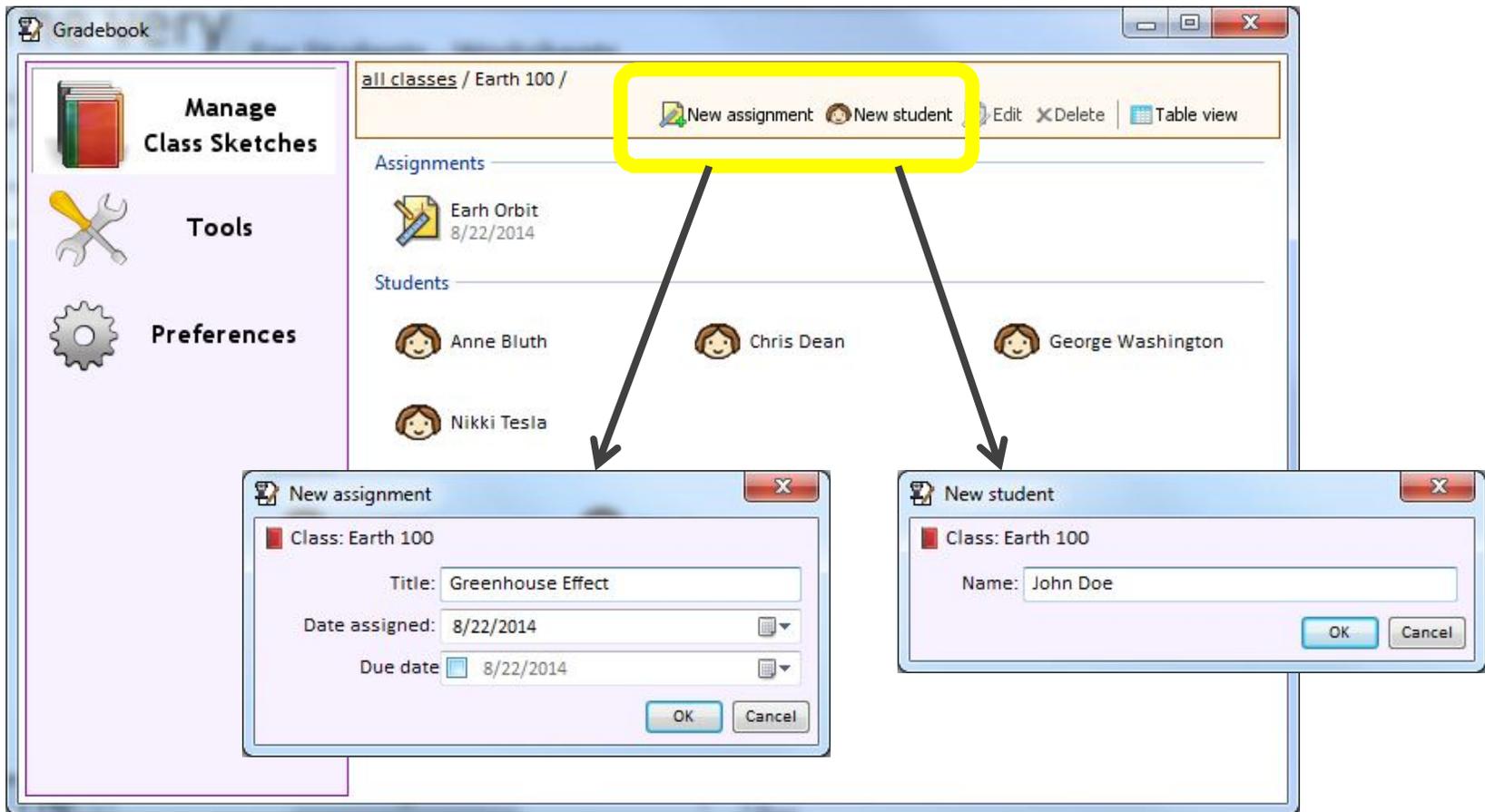
- Try different variations of correct answers
  - Does the tutor view them all as correct?
- Try different error types
  - Missing glyphs
  - Extra glyphs
  - Different spatial arrangements
  - Incorrect conceptual relationships
  - Any other variations you can think of
- Pilot test with friends or small groups of students
- Testing is crucial because students often do things we do not expect!

# Gradebook: adding classes

- Gradebook will initially be empty
- Add classes that you want to keep track of



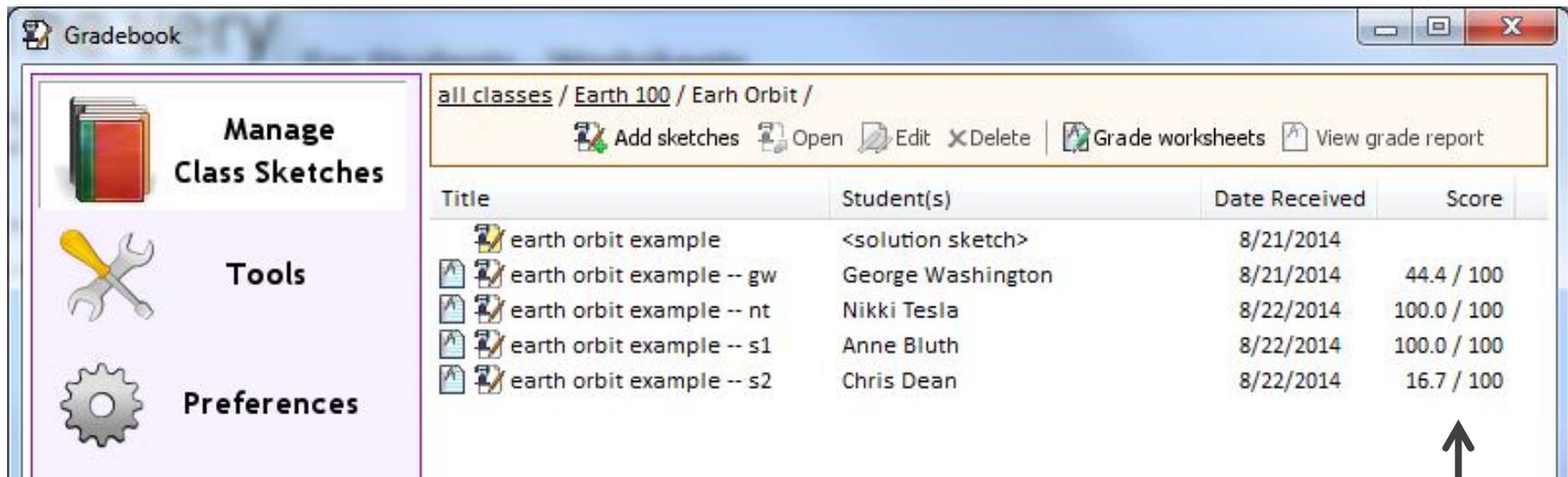
# Gradebook: adding students and assignments



(fictitious students, for privacy)

# Gradebook: managing student sketches

- Double click on assignment to view/add sketches
- Uploading a solution sketch enables **automatic grading**



The screenshot shows the Gradebook application window. On the left is a sidebar with three sections: 'Manage Class Sketches' (with a book icon), 'Tools' (with a wrench and screwdriver icon), and 'Preferences' (with a gear icon). The main area displays a breadcrumb path 'all classes / Earth 100 / Earth Orbit /' and a toolbar with buttons for 'Add sketches', 'Open', 'Edit', 'Delete', 'Grade worksheets', and 'View grade report'. Below the toolbar is a table with the following data:

Title	Student(s)	Date Received	Score
earth orbit example	<solution sketch>	8/21/2014	
earth orbit example -- gw	George Washington	8/21/2014	44.4 / 100
earth orbit example -- nt	Nikki Tesla	8/22/2014	100.0 / 100
earth orbit example -- s1	Anne Bluth	8/22/2014	100.0 / 100
earth orbit example -- s2	Chris Dean	8/22/2014	16.7 / 100

These scores are based on the grading rubrics defined in the worksheet

# Worksheet Studies: Lessons Learned

- New improvements to authoring environment
  - Important sketch facts now viewable as natural language
  - Ability to add domain-specific concepts on a per worksheet basis and query for additional facts on-demand
- New improvements to feedback
  - Feedback at a glance (filling meters) and detailed (text)
  - More detailed feedback options for location-specific items, e.g. "Your right atrium is incorrect."  
→ "Your right atrium is too high."

## Facts Important for Tutoring:

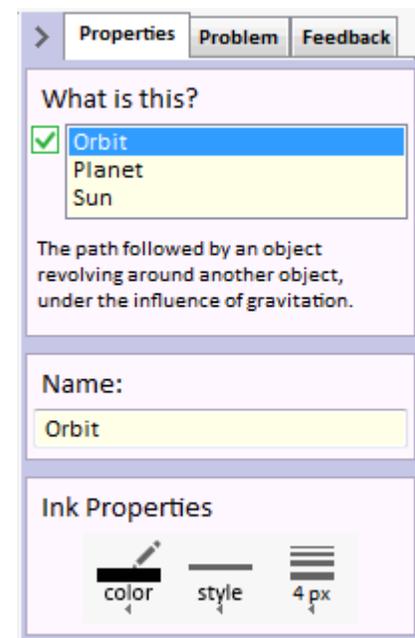
Blood flows from Right atrium to Right ventricle.  
Blood flows from Left atrium to Left ventricle.  
Left atrium is to the right of Right atrium.

The screenshot shows a software interface with three tabs: Properties, Problem, and Feedback. The Feedback tab is active. It displays a section titled "Your Progress" with an "Update" button. Below this, there are several progress indicators:

- "All Glyphs have Labels": A green progress bar with a checkmark icon, indicating 100% completion.
- "Required Glyphs": A yellow progress bar with a dropdown arrow and a warning icon, indicating partial completion.
- "Atria": A yellow progress bar with a dropdown arrow and a warning icon, indicating partial completion.
- "Ventricles": A green progress bar with a checkmark icon, indicating 100% completion.
- "Blood Flow Relations": A yellow progress bar with a dropdown arrow and a warning icon, indicating partial completion.
- "Overall Sketch": A yellow progress bar with a dropdown arrow and a warning icon, indicating partial completion.

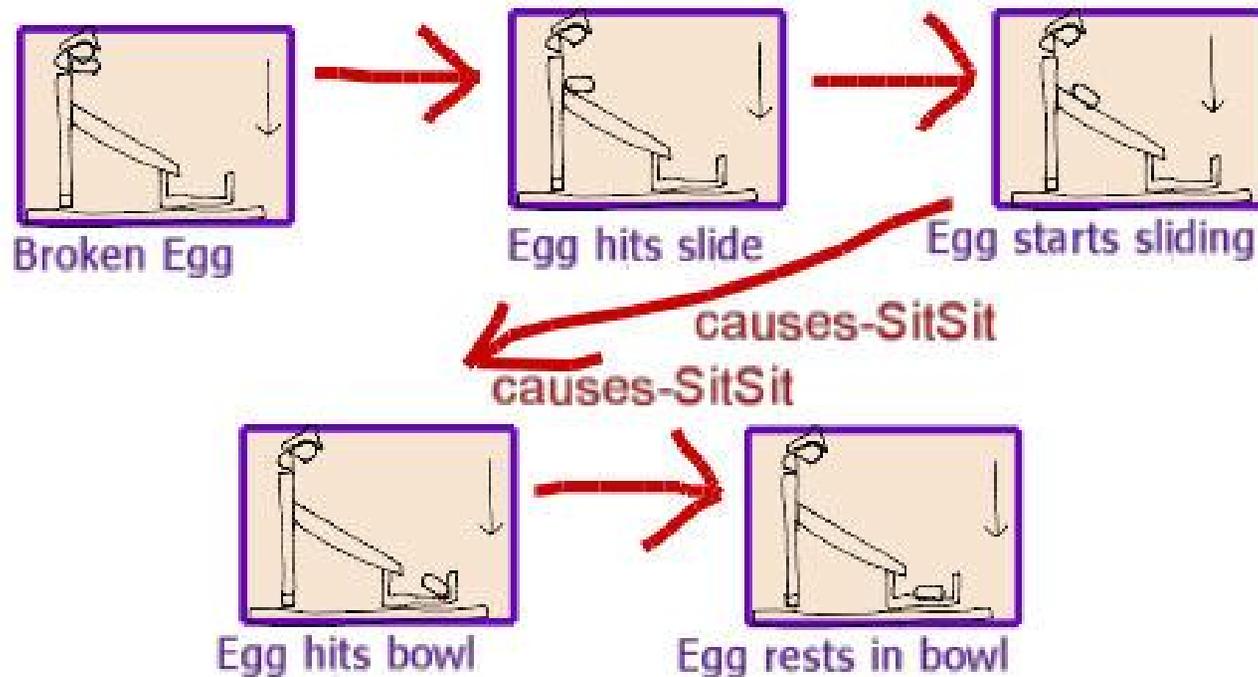
# Worksheet Studies: Lessons Learned

- New User experience improvements
  - Bigger icons, reduced clutter to improve touch friendliness
  - Sidebar instead of pop ups for labeling and feedback improves transparency
- Usage and Usability
  - May be used as homework assignments or in-class and group activities
  - Undergrads: learn software and complete ~2 worksheets in < 1 hour
  - 5<sup>th</sup> graders: positive user experience
    - Likert scale class average: 5.8/7 response to “I liked using CogSketch”



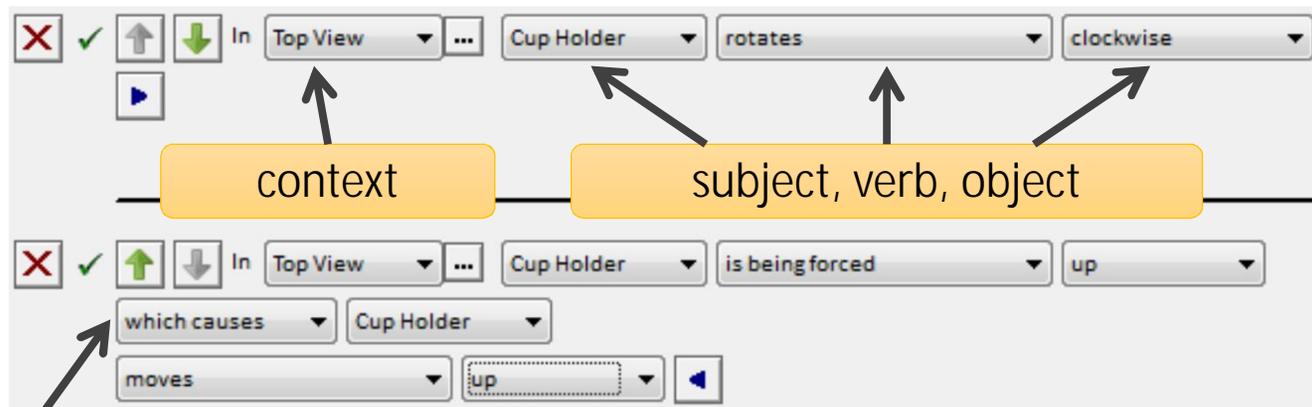
# Supporting Designs with multiple states

- Multiple sketches can depict the design in different operating states
- Student draws arrows between states (e.g. causation), forming a comic graph



# Supporting language-like explanations

- Template-based entry of sentences
- Subject-verb-object form, may be compound



- Coach uses information from sketch *and* language to evaluate the student's explanation

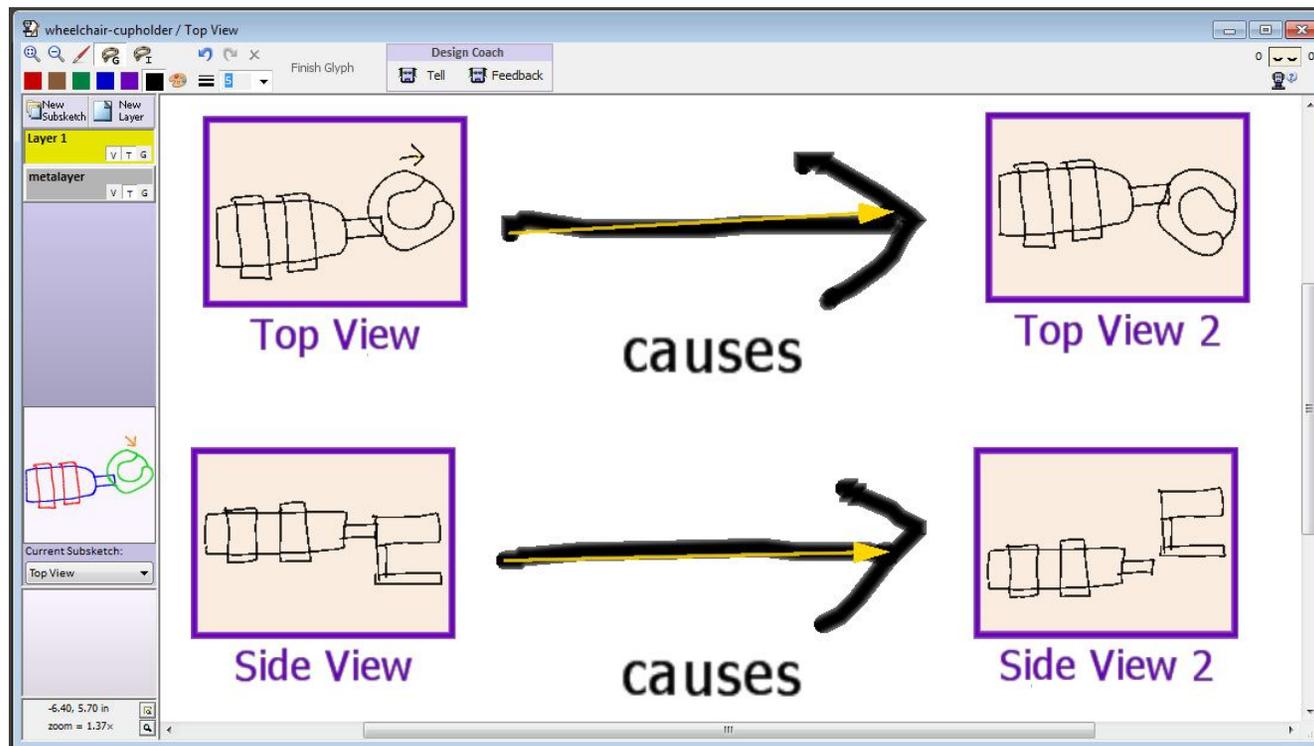
# Design Coach in the Lab and Classroom

- Qualitative physical reasoning expanded to improve coverage of
  - Past design projects [Wetzel & Forbus, 2009]
  - Optional homework assignments from 2010-2011 academic year
  - Mandatory homework assignments from 2011-2012 academic year
- Capabilities now include understanding of
  - Mechanical properties: support forces, motion, springs, gears, pulleys, cords
  - Teleological vocabulary: increasing comfort, containing/holding, attaching/detaching



# Supporting Designs with multiple states

- Multiple sketches can depict the design in different operating states
- Student draws arrows between states (e.g. causation), forming a comic graph



# Analogical Comparison

- Mapping between student and solution sketch
- *Correspondences*: which items match
  - Match constraints: only entities of the same type can match
- *Candidate inferences* (i.e. important differences)
  - True in the base, hypothesized to be true in the target

Base: Solution Sketch	Target: Student Sketch
right-atrium	right-atrium
left-atrium	left-atrium
right-ventricle	right-ventricle
left-ventricle	left-ventricle

# Analogical Comparison

- Mapping between student and solution sketch
- *Correspondences*: which items match
  - Match constraints: only entities of the same type can match
- *Candidate inferences* (i.e. important differences)
  - True in the base, hypothesized to be true in the target

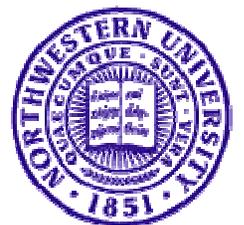
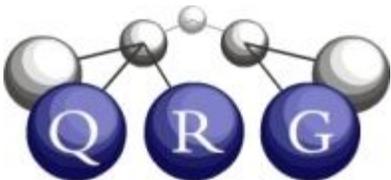
Base: Solution Sketch	Target: Student Sketch
(rightOf right-atrium left-atrium)	(rightOf right-atrium left-atrium)
(above right-atrium right-ventricle)	(above right-atrium right-ventricle)
(bloodFlows left-atrium left-ventricle)	(bloodFlows left-ventricle left-atrium)
(bloodFlows right-atrium right-ventricle)	(bloodFlows right-atrium right-ventricle)

# Advanced Topics

A bluffer's guide to Cyc-style knowledge

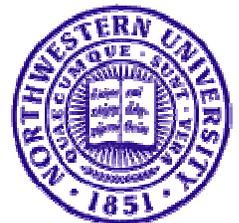
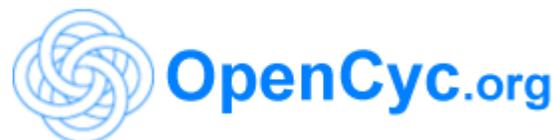
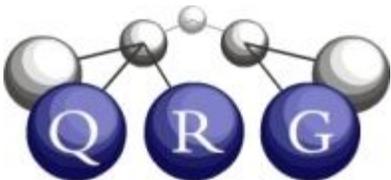
Browsing/adding knowledge in CogSketch  
Working memory and the knowledge base

A KQML API for connecting CogSketch to  
other software



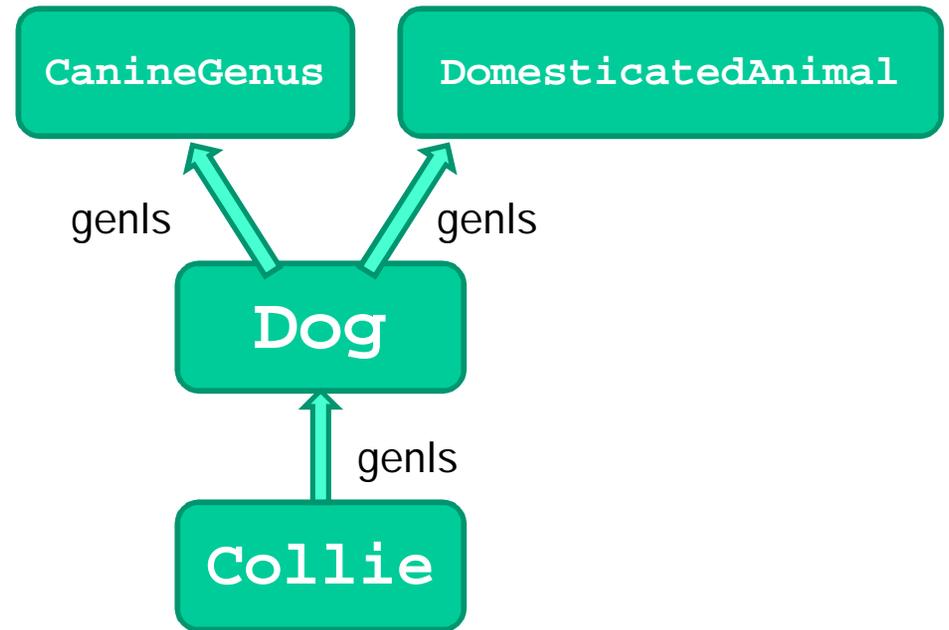
# OpenCyc Knowledge Base

- Cyc = World's largest and most complete general knowledge base
  - Hundreds of thousands of terms
  - Many millions of assertions
  - Rich set of lexical and linguistic linkages to concepts
- OpenCyc = open-source subset of Cyc
  - Freely available
  - Much smaller
- CogSketch uses OpenCyc KB contents
  - Selected a subset relevant for our purposes
  - Added extensions to support new capabilities



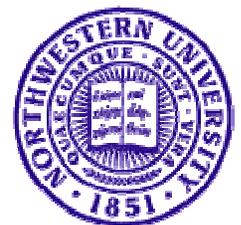
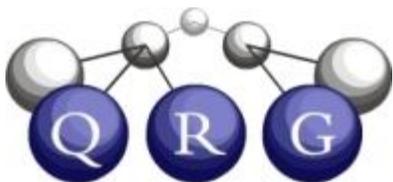
# Collections and Genls

- Concepts and categories in OpenCyc are modeled as *collections*
  - Collection names begin with capital letters
- Collections are related to each other through the *genls hierarchy*



*Everything that is an instance of Collie is also an instance of Dog but not vice versa*

Collie is the collection of all dogs of the breed Collie



# Individuals

- An *individual* is a single thing, not a collection
- Individuals do not have instances—they *are* instances
- Use *isa* to relate an individual to a collection



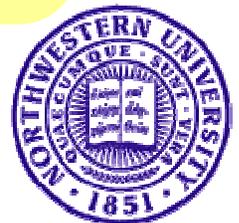
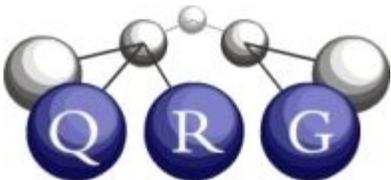
(isa Lassie1 Dog)



(isa Timmy1 MaleChild)

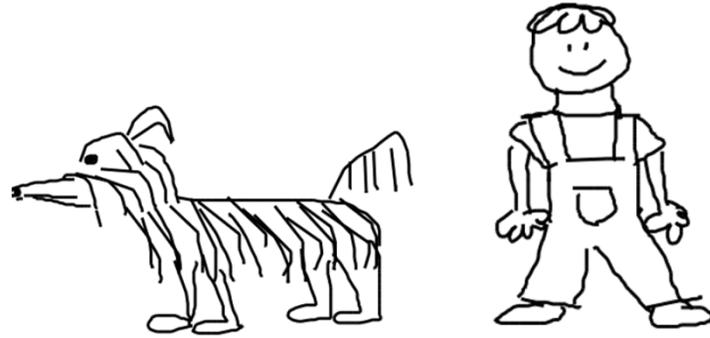
Lassie1 is an *instance* of the *collection* Dog.

Lassie1 is also an *individual*.



# Predicates and genIPreds

- *Predicates* are used to build *sentences*
  - Predicate names begin with lower-case letters
- A sentence built with a predicate is either true or false
- *genIPreds* indicates a hierarchical relationship between predicates



(owns Timmy1 Lassie1)

True

(biologicalRelatives Timmy1  
Lassie1)

False

*Predicates can also relate Collections*

(animalTypeMakesSoundType  
Dog  
BarkingSound)

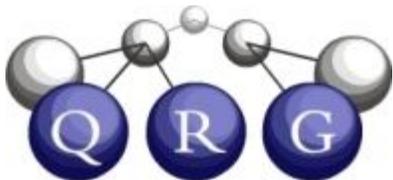
(genIPreds mother  
biologicalRelative)

(disjointWith Cat Dog)



# Arity and Argument Types

- Every predicate has two central features:
  - *Arity*: How many arguments does it require?
  - *Argument types*: What types of arguments does it require?
    - argNisa
    - argNGenl
- Every sentence must be both *semantically* and *syntactically* well-formed



Predicate: owns  
arity: 2  
arg1I sa: SocialBeing  
arg2I sa: SomethingExisting

(owns Timmy1 Lassie1)

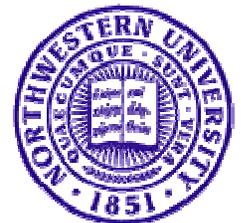
OK!

(owns Timmy1 Lassie1 Rover2)

**Syntactically** poorly-formed

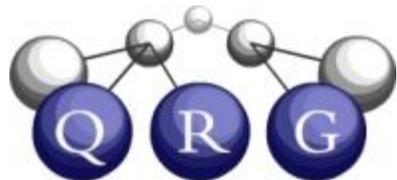
(owns Timmy1 Dog)

**Semantically** poorly-formed



# Microtheories

- The knowledge in OpenCyc is organized into *Microtheories*
- Microtheories can be based on time, source, perspective, ...
- Facts **within** a microtheory must be mutually consistent
- Facts in **different** microtheories may be inconsistent



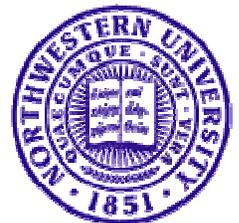
TimmyInWellMT

```
(objectFoundInLocation Timmy1  
OldWell1)  
(isa Lassie1 Dog)  
...
```

Inconsistent but in different  
Microtheories

TimmyEatsDinnerMT

```
(objectFoundInLocation Timmy1  
Home1)  
(isa Lassie1 Dog)  
...
```



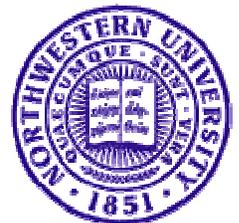
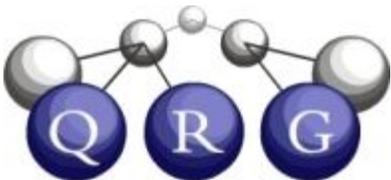
# Using Microtheories

- To make a new microtheory
  - `(isa TimmyInWellMT Microtheory)`
- To relate one microtheory to another
  - `(genlMt TimmyInWellMT LassieMT)`

Every assertion that is true in LassieMT is also true in TimmyInWellMT

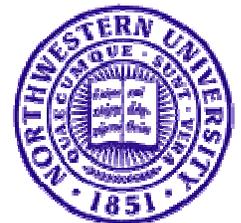
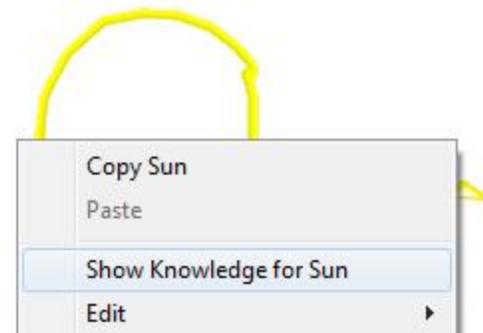
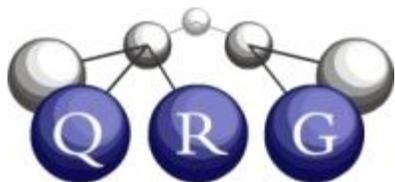
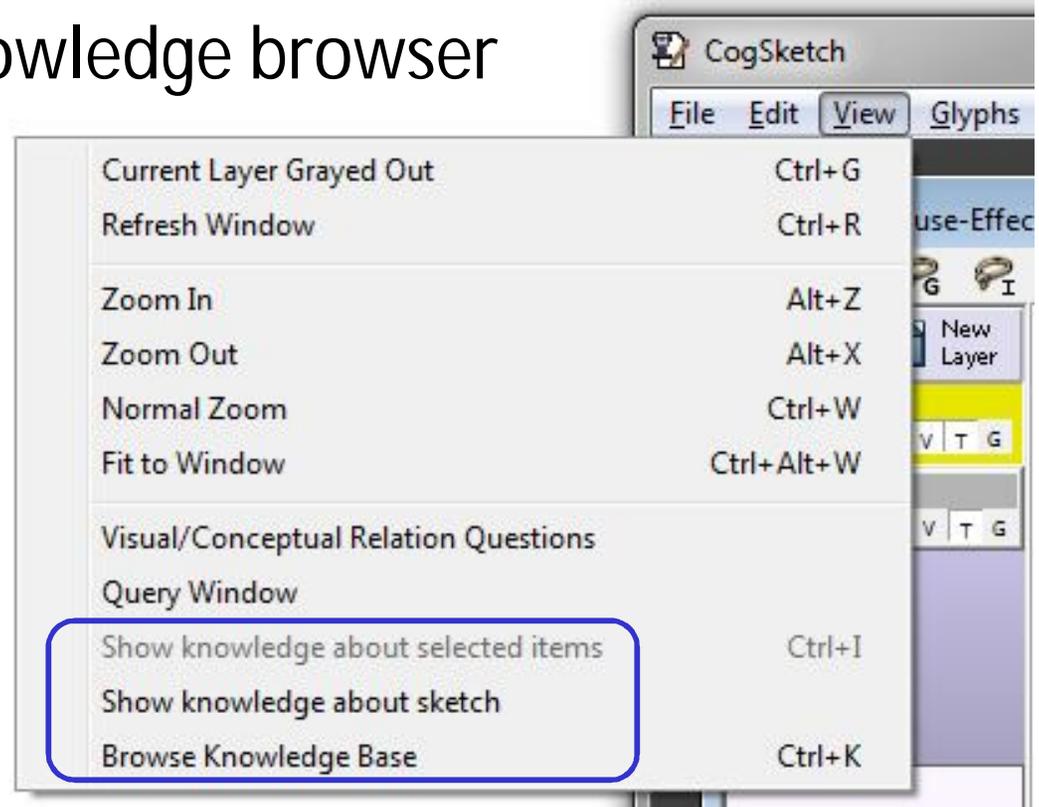
- To make a statement in a microtheory
  - `(ist-Information LassieMT`  
`(isa Lassie1 Dog))`

The assertion `(isa Lassie Dog)` **is true** in the microtheory LassieMT



# Browsing Knowledge in CogSketch

- You can open the knowledge browser from the View menu
- You can see browse knowledge about the whole sketch, selected items, or the whole KB
- You can also right-click on a glyph to see its knowledge



# Knowledge Browser Interface

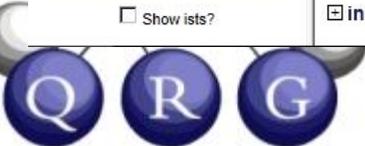
HTML based, Opens in your default web browser

Main pane lists facts about the selected item

Side pane links to sketch, subsketches, layers, and glyphs

The screenshot shows a web browser window with the URL `http://127.0.0.1/rbrowse/ki-frameset.html?id=1`. The interface is divided into several sections:

- Left Sidebar:** Contains a tree view of the workspace. It includes sections for "Subsketch Workspace" (Workspace-3512933206) with layers like "Layer Positional", "Layer Voronoi", and "Layer Layer 1"; "Subsketch Solution" (Solution-3512933211) with layers like "Layer Positional", "Layer Voronoi", and "Layer Layer 1"; and a "Sun" object. At the bottom of the sidebar is a menu with options: "Clarify Glyph Relationships", "Query / WM Fact Edit", "KB Fact Edit", "Analogy", "Refresh Object List", "Browse all WM", "Browse KB", and "Show lists?".
- Top Panel:** Features "Group By" (predicate, none) and "Sort By" (alpha) dropdown menus.
- Main Content Area:** Displays "Sketch 1\_Greenhouse-Effect (Case-3512933206)". A highlighted section titled "Applicable Tutoring Facts" lists facts such as "(emitsLongWaveRadiationTo Object-7577 Object-7578)" with associated "Advice Text" and "Alert Text".
- Bottom Panel:** Lists various fact categories with their counts: "askConceptualForBinaryVisualRelation [14 facts]", "classicWorksheetMatchConstraints [1 fact]", "forceLastModifiedTime [2 facts]", "genIMt [4 facts]", "glyphRepresentsObject [2 facts]", and "inkLastModifiedTime [2 facts]".



# Knowledge Browser Interface

Note that the sketch and each subsketch has its own microtheory

The screenshot displays the Knowledge Browser Interface. On the left, a sidebar lists subsketches under 'Subsketch Workspace' and 'Subsketch Solution'. The main area shows 'SKetch 1\_Greenhouse-Effect (Case-3512933206)' with a green box titled 'Applicable Tutoring Facts' containing three fact entries with advice text. Below this is a list of utility categories like 'askConceptualForBinaryVisualRelation' and 'classicWorksheetMatchConstraints'. At the bottom left, a utility menu includes options like 'Clarify Glyph Relationships' and 'Browse all WM'. A 'Group By' dropdown and 'Apply Grouping/Sorting' button are at the top right.

Special facts may be placed in a colored box at the top for easy access

Useful utilities!



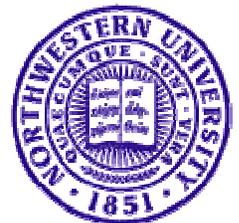
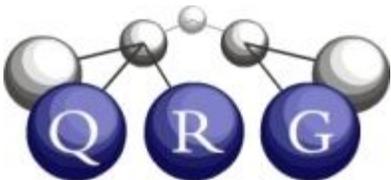
# Working Memory (WM) vs Knowledge Base (KB)

## Working Memory

- Each sketch has one
- Changes as user works with the sketch
- Reflects content of individual sketch

## Knowledge Base

- Each CogSketch installation has one
- Generally not affected by the sketch
- Source of concepts and relations for all sketches



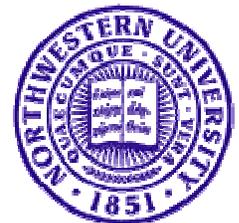
# Knowledge Browser and Working Memory

The screenshot shows a software interface with a menu on the left and a toolbar on the right. The menu items are: Clarify Glyph Relationships, Query / WM Fact Edit, KB Fact Edit, Analogy, Refresh Object List, Browse all WM (with a folder icon), and Browse KB (with a folder icon). Below the menu is a checkbox labeled "Show lists?". The toolbar contains several icons with partial labels: a plus sign in a square followed by "cl", a plus sign in a square followed by "fo", a plus sign in a square followed by "ge", a plus sign in a square followed by "al", and a plus sign in a square followed by "in". Three yellow callout boxes provide descriptions: the first points to the "cl" icon and says "Allows you to add additional visual/conceptual relations to the sketch's WM"; the second points to the "Query / WM Fact Edit" menu item and says "Search for facts/ask if a fact is true, and edit facts in WM"; the third points to the "Browse all WM" menu item and says "Browse the entire WM in one window".

Allows you to add additional visual/conceptual relations to the sketch's WM

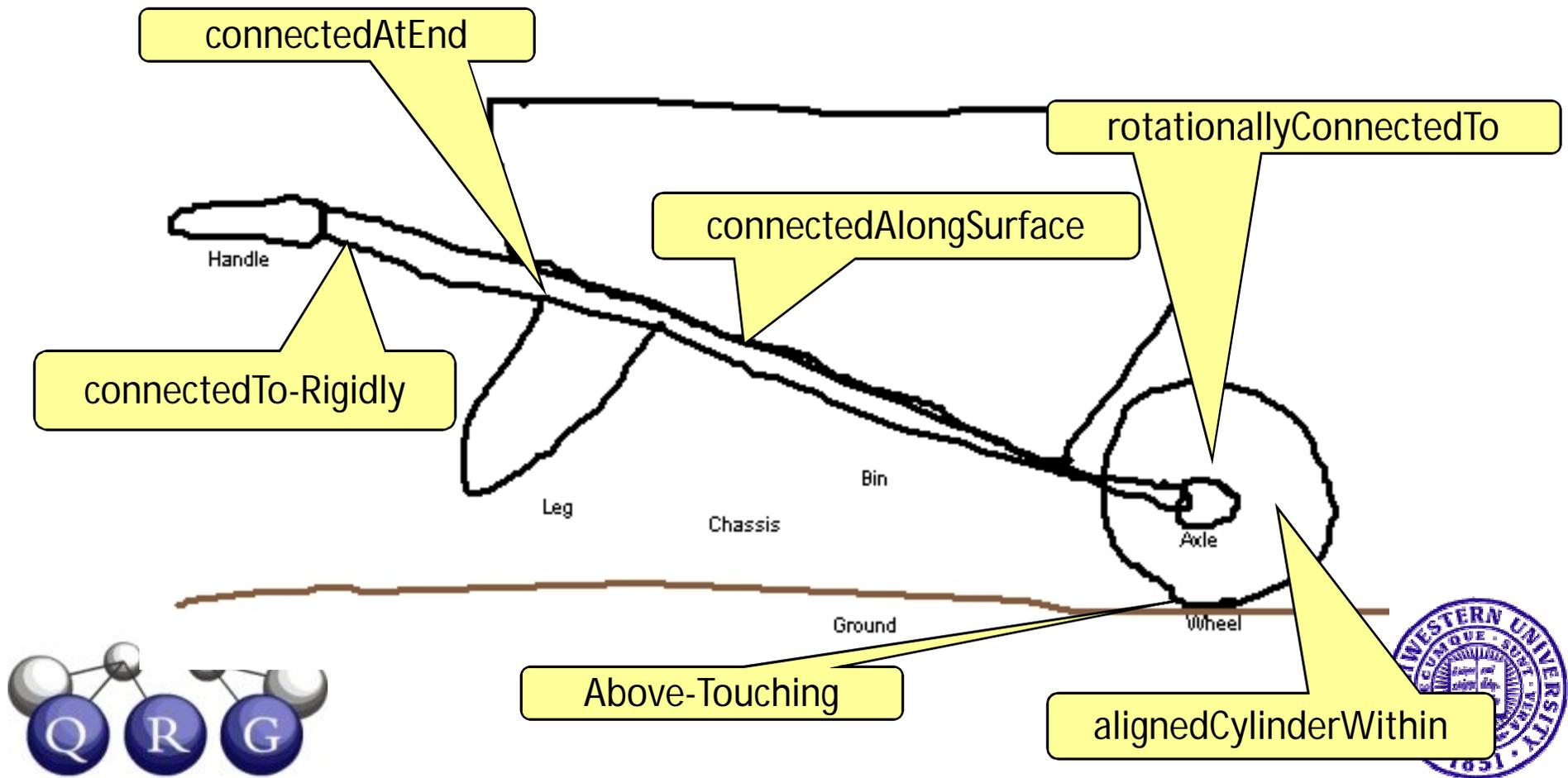
Search for facts/ask if a fact is true, and edit facts in WM

Browse the entire WM in one window



# Visual/Conceptual Relationships

- People use conventions for depicting physical relationships in sketches
- You can tell CogSketch about your assumptions



# Example: Shopping Cart

## (GlyphFn Object-147 User-Drawn-Sketch-Layer-225)

human-readable namestring: front wheel  
glyph represents: Object-147

### isa [6 facts]

? A (isa Object-147 Entity)

? A (isa Object-147 Wheel)

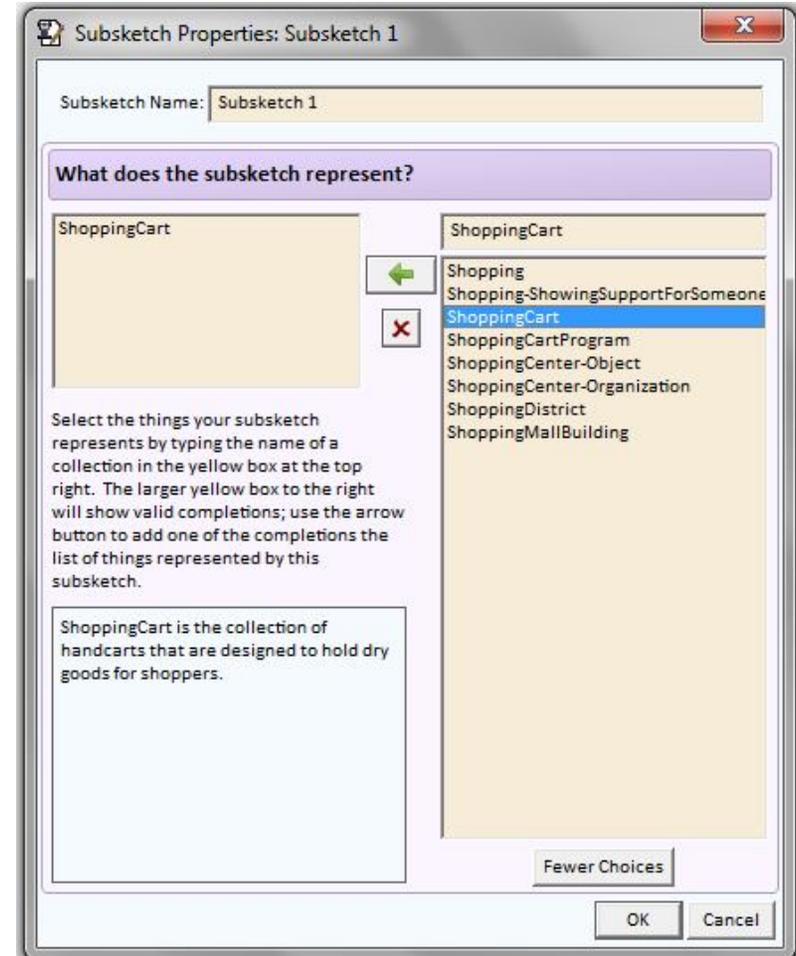
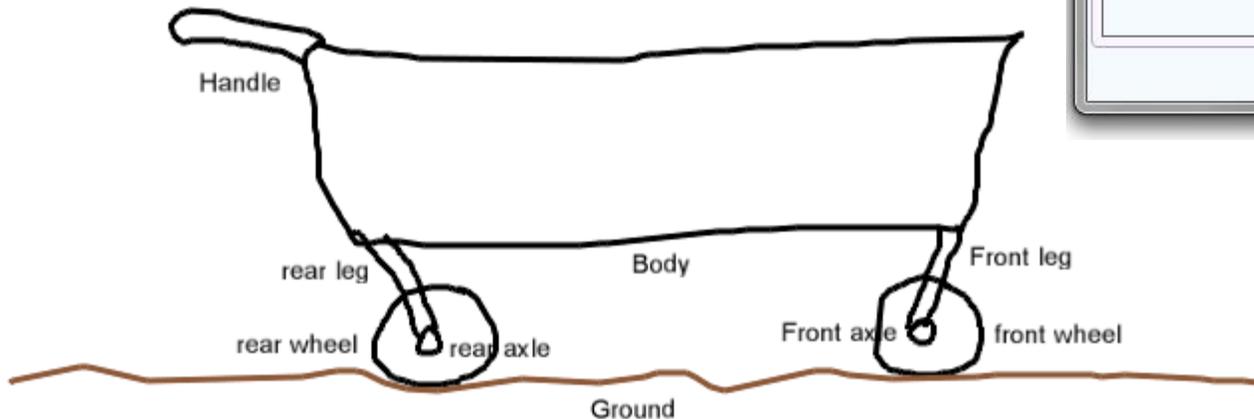
### spatiallyIntersects [4 facts]

? A (spatiallyIntersects  
(GlyphFn Object-147 User-Drawn-Sketch-Layer-225)  
(GlyphFn Object-150 User-Drawn-Sketch-Layer-225))

? A (spatiallyIntersects  
(GlyphFn Object-147 User-Drawn-Sketch-Layer-225)  
(GlyphFn Object-153 User-Drawn-Sketch-Layer-225))

? A (spatiallyIntersects  
(GlyphFn Object-150 User-Drawn-Sketch-Layer-225)  
(GlyphFn Object-147 User-Drawn-Sketch-Layer-225))

? A (spatiallyIntersects  
(GlyphFn Object-153 User-Drawn-Sketch-Layer-225)  
(GlyphFn Object-147 User-Drawn-Sketch-Layer-225))



# Providing Visual/Conceptual Relations

## Bundle Shopping Cart Anatomy:

Conceptual relationships between Body and Front leg:

### User supplied relationship

Which of the following best describes the relationship between Body and Front leg?

(connectedAtEnd Front leg Body)

Conceptual relationships between Body and Handle:

### User supplied relationship

Which of the following best describes the relationship between Body and Handle?

(connectedAtEnd Handle Body)

Conceptual relationships between Ground and front wheel:

### User supplied relationship

Which of the following best describes the relationship between Ground and front wheel?

(above-Touching front wheel Ground)

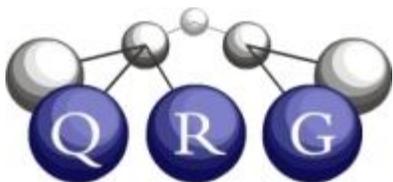
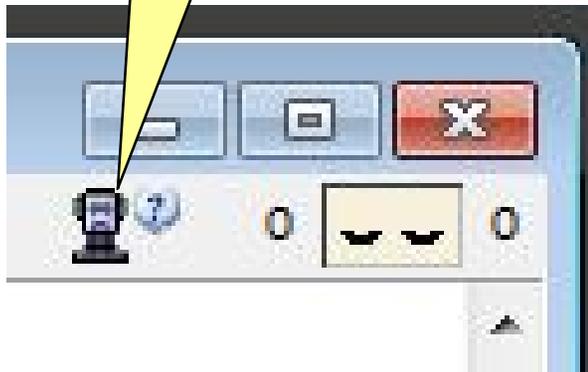
Conceptual relationships between Front axle and front wheel:

### User supplied relationship

Which of the following best describes the relationship between Front axle and front wheel?

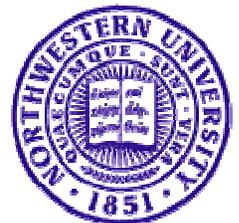
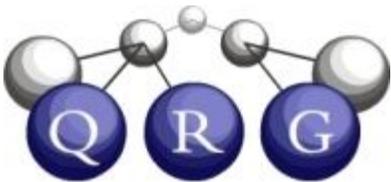
(alignedCylinderWithin Front axle front wheel)

Use this button to bring up web interface on selected bundle



# How Visual/Conceptual Relations are Hypothesized

- Qualitative topology used to suggest initial candidates
  - (`insideInSketch o1 o2`) if (`glyph o1`) is inside (`glyph o2`)
  - (`atOrOverlapsInSketch o1 o2`) if (`glyph o1`) touches or overlaps (`glyph o2`)
- Possible specializations filtered by argument type relationships
- You can choose more specialized relationship if desired
- Not an easy problem
  - Worst case: 150 possibilities for `insideInSketch`, 204 for `atOrOverlapsInSketch`, with ResearchCyc KB
  - For one corpus of 34 sketches:
    - Mean # questions/sketch = 4
    - Mean # candidates to consider per question = 122



# Example: Front Wheel/Axle

Conceptual relationships between Front axle and front wheel:

**User supplied relationship**

Which of the following best describes the relationship between Front axle and front wheel?

--

(alignedCylinderWithin Front axle front wheel)

(artifactFoundInLocation Front axle front wheel)

(commerciallyUsefulParts Front axle front wheel)

(connectedToInside Front axle front wheel)

(constituents Front axle front wheel)

(cospatial Front axle front wheel)

(covers-Baglike Front axle front wheel)

(embeddedCylinderInSheet Front axle front wheel)

(entirePortion Front axle front wheel)

(externalParts Front axle front wheel)

(hasStoredInside Front axle front wheel)

(in-Among Front axle front wheel)

(in-ContClosed Front axle front wheel)

(in-ContCompletely Front axle front wheel)

(in-ContFullOf Front axle front wheel)

(in-ContGeneric Front axle front wheel)

(in-ContOpen Front axle front wheel)

(in-Rooted Front axle front wheel)

(in-Snugly Front axle front wheel)

(inRegion Front axle front wheel)

(inertIngredients Front axle front wheel)

(ingredients-Constituent Front axle front wheel)

(ingredients-Separable Front axle front wheel)

(internalParts Front axle front wheel)

(internalSubRegions Front axle front wheel)

(localityOfObject Front axle front wheel)

(mainConstituent Front axle front wheel)

(objectFoundInLocation Front axle front wheel)

(objectSides Front axle front wheel)

(physicalDecompositions Front axle front wheel)

(physicalParts Front axle front wheel)

(physicalParts-Separated Front axle front wheel)

(physicalPortions Front axle front wheel)

(physicallyContains Front axle front wheel)

(pigments Front axle front wheel)

(pluggedInto Front axle front wheel)

(properPhysicalDecompositions Front axle front wheel)

(properPhysicalParts Front axle front wheel)

(properlySpatiallySubsumes Front axle front wheel)

(properlySpatiallySubsumes-Nontangential Front axle front wheel)

(properlySpatiallySubsumes-Tangential Front axle front wheel)

(protectiveContains Front axle front wheel)

(protrudesInto Front axle front wheel)

(screwedIn Front axle front wheel)

(spans-Bridgelike Front axle front wheel)

(spatiallyContains Front axle front wheel)

(spatiallyIncludes Front axle front wheel)

(spatiallySubsumes Front axle front wheel)

(sticksInto Front axle front wheel)

(sticksInto-2D Front axle front wheel)

(subRegions Front axle front wheel)

(surfaceParts Front axle front wheel)

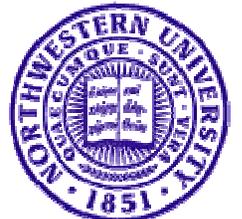
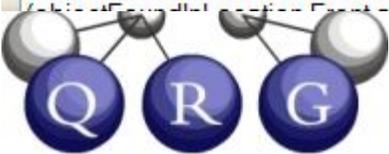
(surrounds-3D Front axle front wheel)

(surroundsCompletely Front axle front wheel)

and front wheel?

and front wheel?

and rear wheel?



# Querying from Working Memory

Query / WM Fact Edit

Use a "?" before an argument to make it a query variable, e.g. "?x" or "?object"

Enter a fact or query here:

(touchesDirectly ?x ?y)

Microtheory to ask/query/tell/untell in. Defaults to everything. You can choose a subsketch's microtheory (e.g. Bcase-xxxxxxx), for example

Can query WM, KB, or both

Context: EverythingPSC

Facts: all

"query" will use rules in WM and KB

"tell" adds an assumption into WM

"ask" is faster but doesn't use inference

Allow microtheory inheritance? (env)

Allow gens inferencing? (transitive)

Allow other kinds of inference? (infer)

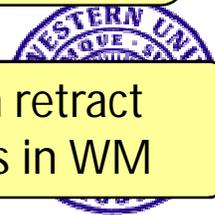
Query using fire:ask

Query using fire:query

Tell

Untell

"untell" can retract assumptions in WM



# Can Get Answers

## Query / WM Fact Edit

```
(touchesDirectly ?x ?y)  
action = ask  
context = EverythingPSC; facts = all, env, infer
```

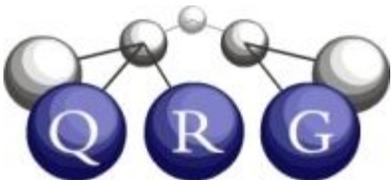
### Answers:

#### in EverythingPSC:

```
? A (touchesDirectly Object-43 Object-44)  
? A (touchesDirectly Object-43 Object-46)  
? A (touchesDirectly Object-43 Object-51)  
? A (touchesDirectly Object-44 Object-43)  
? A (touchesDirectly Object-44 Object-51)  
? A (touchesDirectly Object-44 Object-68)  
? A (touchesDirectly Object-46 Object-43)
```

Click "?" to  
see why fact  
is believed

Click "A" to see the list of  
underlying assumptions



# Can Drill Down for Reasons

Drill down using the "?" icon

Touches is believed because of this rcc8 relation and this rule in the KB

Justifications		
(touchesDirectly Object-43 Object-44)		

The above expression is true because of the following:

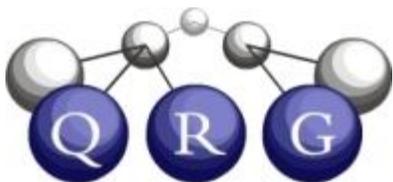
?	A	(hasRCC8Relation (GlyphFn Object-43 User-Drawn-Sketch-Layer-13) (GlyphFn Object-44 User-Drawn-Sketch-Layer-13) rcc8-P0)	[true]
?		(inKB (ist-Information NuSketchQMRulesMt (<== (touchesDirectly ?obj1 ?obj2) (wmOnly (hasRCC8Relation (GlyphFn ?obj1 ?sketch-layer) (GlyphFn ?obj2 ?sketch-layer) rcc8-P0))))))	[true]

It is true via:

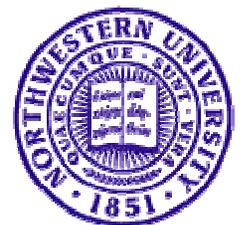
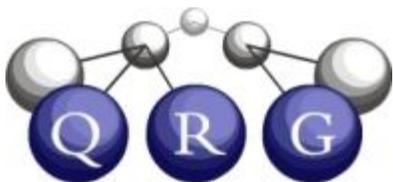
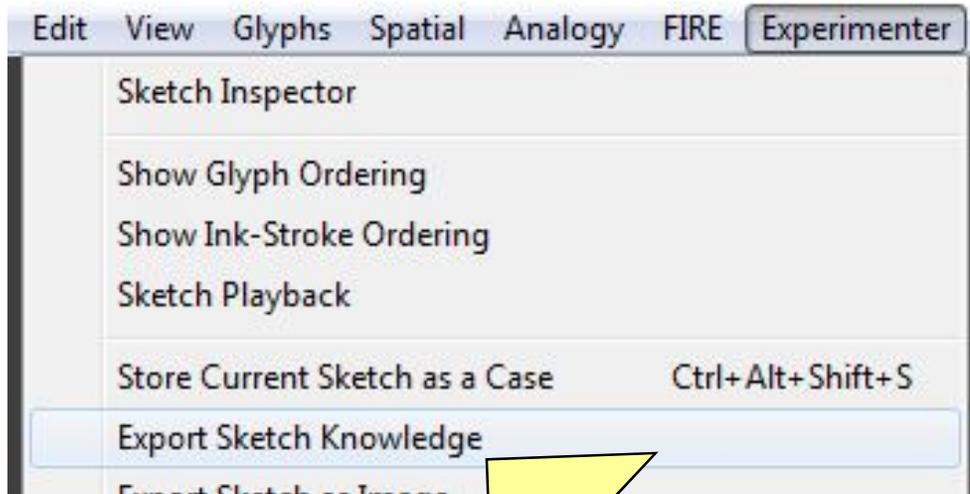
```
(:implied-by
 (:implies
 (:and (inKB (ist-Information NuSketchQMRulesMt
 (<== (touchesDirectly ?obj1 ?obj2)
 (wmOnly
 (hasRCC8Relation
 (GlyphFn ?obj1 ?sketch-layer)
 (GlyphFn ?obj2 ?sketch-layer)
 rcc8-P0))))))
 (ist-Information BCase-3487440799
 (hasRCC8Relation
 (GlyphFn Object-43 User-Drawn-Sketch-Layer-13)
 (GlyphFn Object-44 User-Drawn-Sketch-Layer-13)
 rcc8-P0)))
 (ist-Information BCase-3487440799
 (touchesDirectly Object-43 Object-44)))
 :bc-justify-clause-bc-result)
```

Direct Consequences:

?	A	(touchesDirectly Object-43 Object-44)	[true]
---	---	---------------------------------------	--------



# Exporting Knowledge to Files



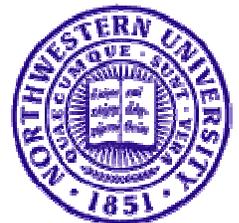
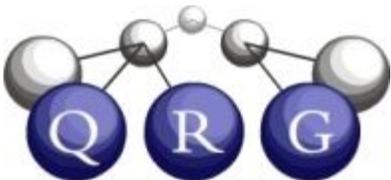
# MELD format files

- Similar to Cyc KE format

```
;; constant: Case-3429195339.  
;; in Mt: BaseKB.  
(isa Case-3429195339 Microtheory)  
(isa Case-3429195339 COASpecificationMicrotheory)  
(genlMt Case-3429195339 BaseKB)
```

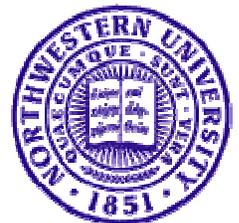
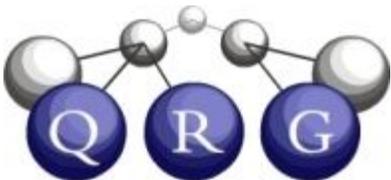
```
;; constant: BCase-3429195452.  
;; in Mt: BaseKB.  
(isa BCase-3429195452 Microtheory)  
(isa BCase-3429195452 COASpecificationMicrotheory)  
(genlMt BCase-3429195452 Case-3429195339)
```

```
;; Default Mt: Case-3429195339.
```



# Working Memory - Summary

- You can add visual-conceptual relations to a sketch's knowledge
  - Useful for reasoning about what the sketch depicts
- You can view the facts about a sketch and all its parts with the knowledge browser
  - Drill down and see why those facts are believed
  - Useful for debugging
- You can make queries to infer new facts or look up old ones
- You can tell/untell some facts into WM
- You can export facts from your sketches to files



# Working with the Knowledge Base

Object-1378

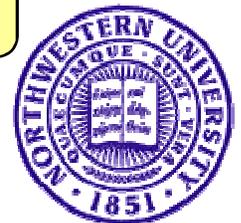
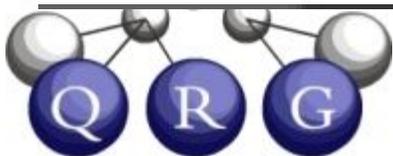
- Clarify Glyph Relationships
- Query / WM Fact Edit
- KB Fact Edit
- Analogy
- Refresh Object List
- Browse all WM
- Browse KB

Show lists?

- cli
- fo
- ge
- gl
- in

Retrieve/store facts in the KB

Browse the entire KB



# Example: Browsing

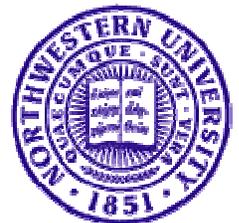
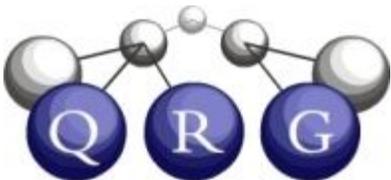
- Let's look for other relationships involving rotation with the KB browser

rotat search

---

## Possible matches for "rotat":

- Rotataion-None (Collection)
- RotatedShape-180 (Collection)
- RotatedShape-45 (Collection)
- RotatedShape-90 (Collection)
- RotatedShape-None (Collection)
- Rotation-135 (Collection)
- Rotation-180 (Collection)
- Rotation-45 (Collection)
- Rotation-90 (Collection)
- Rotation-Clockwise135 (Collection)
- Rotation-Clockwise45 (Collection)
- Rotation-Clockwise90 (Collection)
- Rotation-CounterClockwise135 (Collection)



## rotationallyConnectedTo

- all genIPreds
- all specPreds
- all references

## rotationallyConnectedTo [type = Relation]:

comment: A ConnectionPredicate (q.v.) and thus a specialization of connectedTo (q.v.).  
(rotationallyConnectedTo OBJ1 OBJ2) means that OBJ1 and OBJ2 are connected in such a way that rotational motion, and only rotational motion, can happen between them. The range of rotational motion possible might be full or partial. Non-rotational movement between two rotationally connected objects can occur only if the connection is broken, deformed, or disassembled. If OBJ1 and OBJ2 do rotate relative to one another, then this may be due to sliding of their surfaces, articulation of some joint part, or deformation of OBJ1 or OBJ2 (so long as that deformation only allows rotation between OBJ1 and OBJ2). Positive examples: Femurs are rotationally connected to hips, doors are rotationally connected to door frames, doorknobs are rotationally connected to doors, and propellers are rotationally connected to airplanes; in computer trackballs the ball is rotationally connected to the housing. Also a book cover is rotationally connected to its binding (but flapHingedTo is even more appropriate for describing such a connection because it is more specific). Negative examples: a planet orbiting a star (they are not connected; cf. MovingInACircle ) and a toothpick stuck in a person's leg (although elastic deformation of flesh allows there to be rotational motion between toothpick and leg, it also may allow a small amount of translational motion to occur between them; in-Lodged is more appropriate for describing this case).

isa:

**in UniversalVocabularyMt:** ConnectionPredicate , IrreflexiveBinaryPredicate , SymmetricBinaryPredicate

**in TopicMt:** Connections-Spatial-Topic

arity: 2

arg1Isa: SolidTangibleThing

arg2Isa: SolidTangibleThing

genIPreds:

**in BaseKB:** rotationallyConnectedTo

**in UniversalVocabularyMt:** connectedTo

specPreds:

**in UniversalVocabularyMt:** connectedByBeltTo , hingedTo , screwedIn

# Editing the KB through the Knowledge Browser

KB Fact Edit

Enter a fact or query here:

(isa Snoopy ?x)

Microtheory to query in.  
Defaults to everything--this can  
be a poor choice because the  
KB is very large!

Context: EverythingPSC

Retrieve

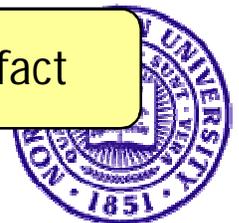
Store

Forget

"Retrieve" looks up facts

"Store" adds fact

"Forget" removes fact



# Knowledge Base Query Result

## KB Fact Edit

```
(isa Snoopy ?x)  
context = EverythingPSC  
action = retrieve
```

### Results:

#### in UniversalVocabularyMt:

```
(isa Snoopy Individual)
```

#### in FictionalWorksMt:

```
(isa Snoopy FictionalCharacter)
```

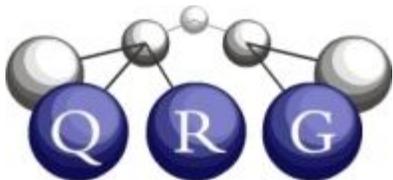
#### in MassMediaDataMt:

```
(isa Snoopy CartoonCharacter)
```

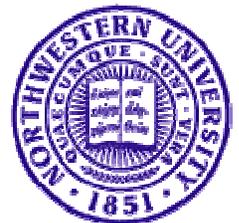
#### in ThePeanutsCartoonMt:

```
(isa Snoopy Beagle) (isa Snoopy MaleAnimal)
```

Note there is no option to drill down for justifications; these are simply facts retrieved from the KB

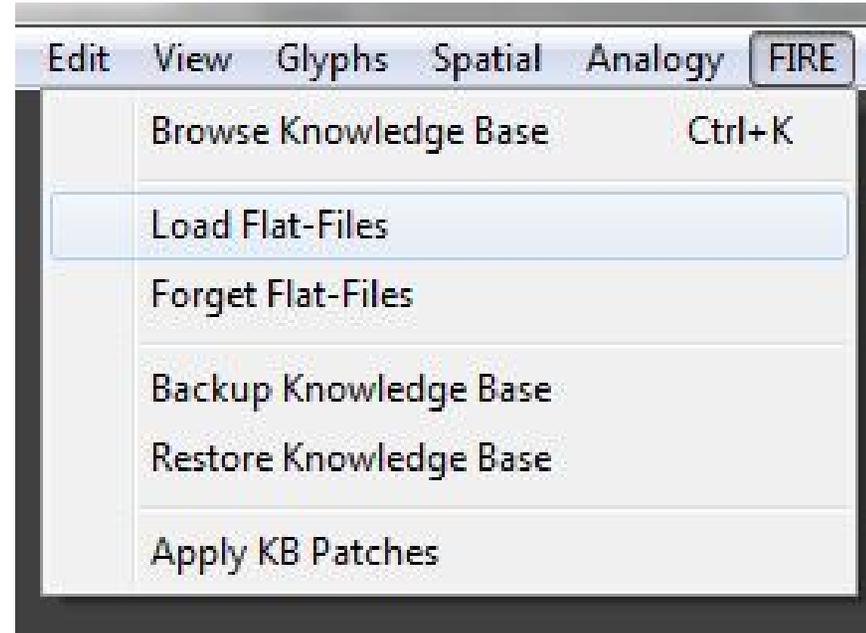


[Return to Fact Edit Page](#)

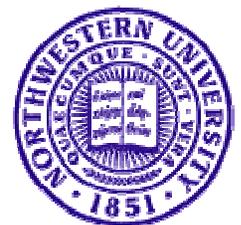
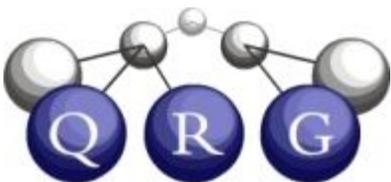


# Extending the Knowledge Base

- OpenCyc has a lot of knowledge ... but it might not have everything you need
- You add knowledge using a .meld file
- Create using your favorite text editor



Hint: Use an editor that matches parentheses, such as emacs!



# Example: A Simple Flat-File

`(in-microtheory TimmyInWellMT) ;;` *Tells file loader what microtheory to use. All forms after this command are facts for that microtheory.*

`(isa Lassie1 Dog)`

`(isa Timmy1 MaleChild)`

`(isa OldWell1 Well)`

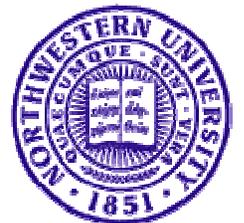
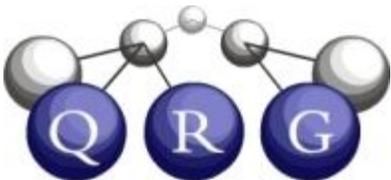
`(owns Timmy1 Lassie1)`

`(objectInLocation Timmy1 OldWell1)`

`(isa LassieGetHelp RescuingSomeone)`

`(performedBy LassieGetHelp Lassie1)`

`(beneficiary LassieGetHelp Timmy1)`

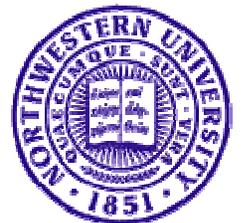
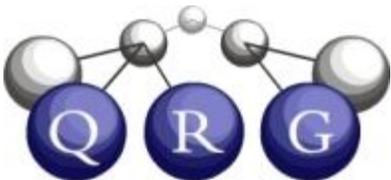


# Adding a Collection

To add a collection you need at least three things:

1. A statement that it is a Collection
2. A gens statement
3. A comment describing the collection

```
(isa Firefly Collection)  
(gens Firefly Insect)  
(comment Firefly  
  "the collection of all  
  insects that having  
  glowing posteriors")
```

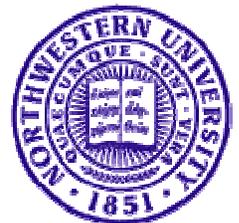
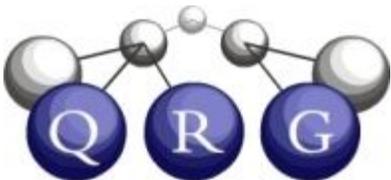


# Adding a Relation

To add a relation you need at least four things:

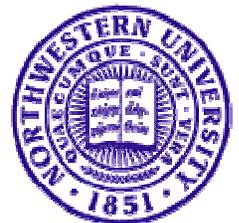
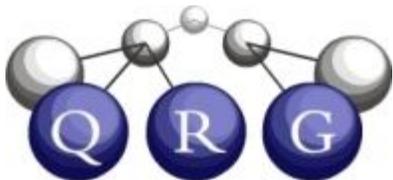
1. A statement that it isa Relation
2. An arity statement
3. ArgIsa statements
4. A comment describing the relation

```
(isa aboveGrazingLine
      Relation)
(arity aboveGrazingLine 2)
(arg1Isa aboveGrazingLine
        NuSketchGlyph)
(arg2Isa aboveGrazingLine
        NuSketchGlyph)
(comment aboveGrazingLine
        "the figure object
        represented by the glyph in
        arg1 is above the grazing
        line created by the ground
        object represented by the
        glyph in arg2")
```



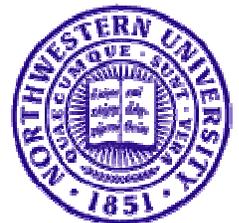
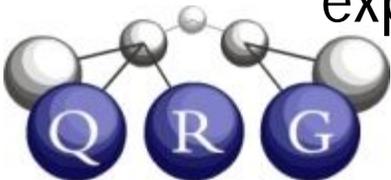
# Using Your New KB entries in CogSketch

- Your new collections
  - Can be used in conceptual labeling
  - Can be used to constrain arguments to relations
- Your new relations
  - Can be used in worksheets
  - Can show up as hypothesized visual/conceptual relationship questions, if you weave them into the `gen1Preds` hierarchy correctly
  - Can be used for your own reasoning, if you add Horn clause axioms involving them also
    - Via browser query window, or API calls
    - Documentation on doing this is in progress



# Knowledge Base - Summary

- You can browse for existing collections and relations
- You can retrieve facts from the KB with the knowledge browser
- You can add your own collections and relations
  - You can store individual facts using the knowledge browser
  - You can add a large amount of facts at one time by making a flat file and importing it
  - New collections and relations can be used in worksheets, cognitive simulations, and other experiments

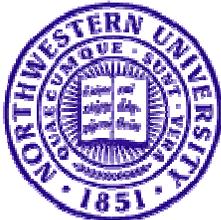
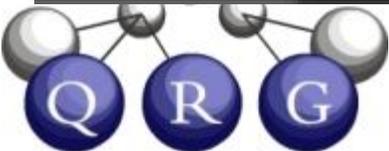


# Browsing Analogies

Object-1378

Clarify Glyph Relationships	+ cli
Query / WM Fact Edit	
KB Fact Edit	+ fo
<b>Analogy</b>	+ ge
Refresh Object List	
Browse all WM 	+ gl
Browse KB 	
<input type="checkbox"/> Show lists?	+ in

Displays recent analogies



# Analogy Browser

Clicking an SME opens it so you can view the list of mappings

## Reasoner Analogy Source

### Recorded SMEs:

- SME #1, version 0

### Cached DGroups:

- (CogSketchTutorBundleCaseFn Workspace-3512933206 Subsketch-8) context = Workspace-3512933206
- (CogSketchTutorBundleCaseFn Solution-3512933211 Subsketch-9) context = Workspace-3512933206

Clear Analogy Source

## SME #1

**Base:** (CogSketchTutorBundleCaseFn Solution-3512933211 Subsketch-9)

**Target:** (CogSketchTutorBundleCaseFn Workspace-3512933206 Subsketch-8)

### Match Constraints

- (requireWithinPartitionCorrespondences Atmosphere)
- (requireWithinPartitionCorrespondences Planet)
- (requireWithinPartitionCorrespondences Relation)
- (requireWithinPartitionCorrespondences Sun)

Generally the mapping with the highest score is used for things like tutoring advice.

Mapping	Score	# MHs	# CIs
<a href="#">Mapping 249</a>	0.6245	86	6
<a href="#">Mapping 330</a>	0.5800	73	13
<a href="#">Mapping 316</a>	0.5320	63	21

- [49 entity correspondences](#)
- [212 expression correspondences](#)
- [27 functor correspondences](#)



# Analogy Browser

Mapping 249

SME #1

Score: 0.6245

Base: (CogSketchTutorBundleCaseFn Solution-3512933211 Subsketch-9)

Target: (CogSketchTutorBundleCaseFn Workspace-3512933206 Subsketch-8)

Support	Base Item	Target Item	MH Score
★ (13)	* Object-7578	* Object-7652	0.1640
★ (12)	* Object-7591	* Object-7654	0.0720
★ (10)	* Object-7592	* Object-7655	0.0560
★ (9)	* Object-7577	* Object-7651	0.1520
★ (7)	* Object-7579	* Object-7653	0.0400
★ (6)	* (ContainedGlyphGroupFn Object-7578 (TheList Object-7577 Object-7591 Object-7592))	* (ContainedGlyphGroupFn Object-7652 (TheList Object-7651 Object-7654 Object-7655))	0.0520
★ (4)	* Object-7576	* Object-7650	0.0200
★ (4)	* User-Drawn-Sketch-Layer-429	* User-Drawn-Sketch-Layer-424	0.0160
★ (1)	* (ConnectedGlyphGroupFn (TheList Object-7577 Object-7578 Object-7579 Object-7591 Object-7592))	* (ConnectedGlyphGroupFn (TheList Object-7651 Object-7654 Object-7655))	0.0040
★ (1)	* LookingFromSide-SubSketch	* LookingFromSide-SubSketch	0.0080
★ (1)	* PhysicalView-SubSketch	* PhysicalView-SubSketch	0.0080

- [6 candidate inferences](#)
- No reverse candidate inferences
- [49 expression correspondences](#)
- [26 functor correspondences](#)

Legend:

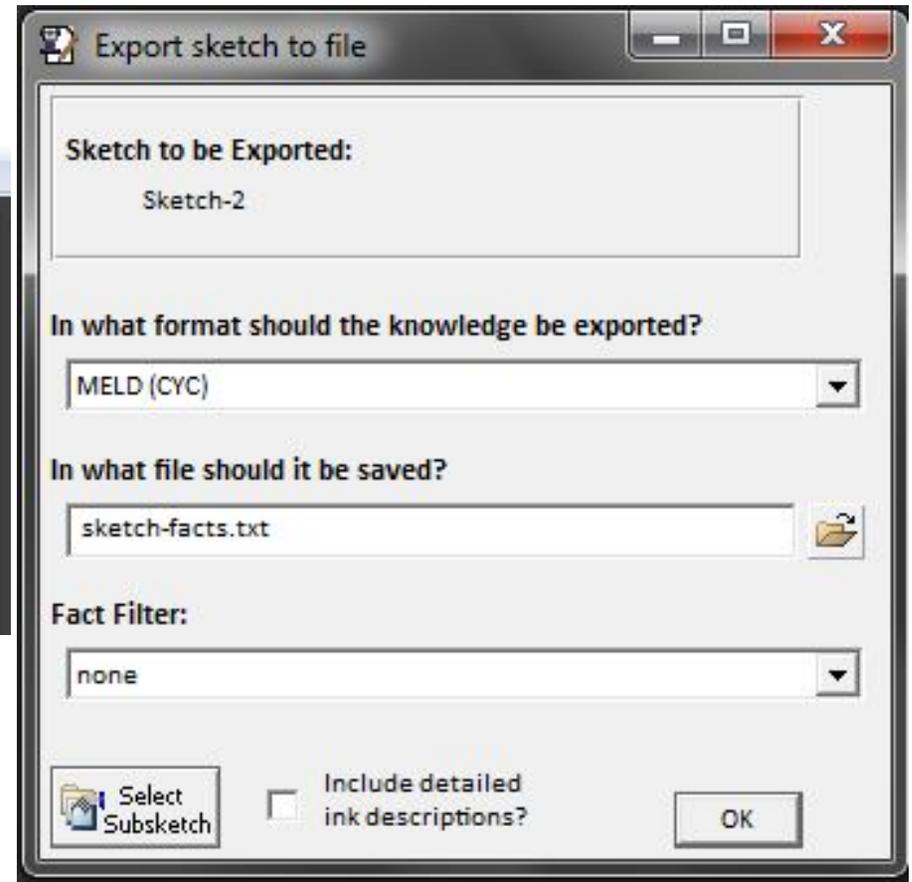
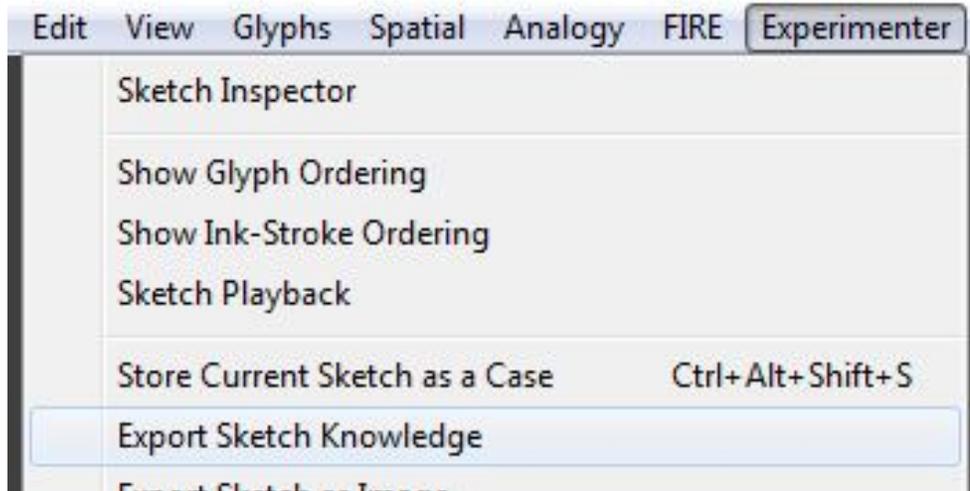
★ = Match Hypothesis Details   \* = Expression Details   ? = Candidate Inference Evaluation

Mapping view shows what was matched with what between subsketches

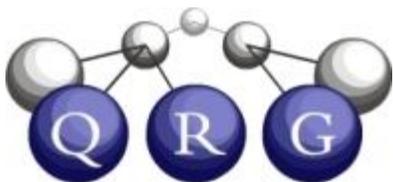


Reminder – you can export knowledge for use in other systems

# CogSketch as a Module

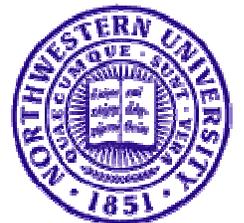
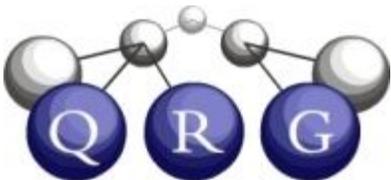


Or access CogSketch directly using the KQML API



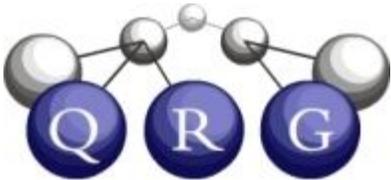
# CogSketch API

- Allows you to access CogSketch from code
- Socket-based, using KQML messages
- Documentation and sample client provided with CogSketch executable
- Suggestions for how we should extend the API to make it more useful for you are very welcome!



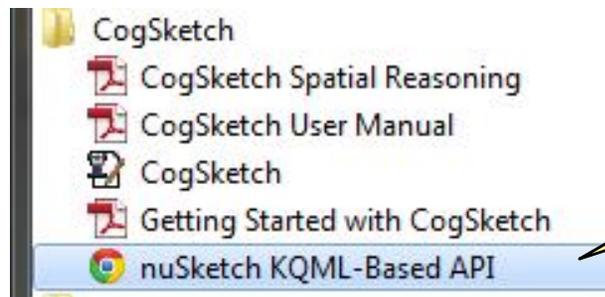
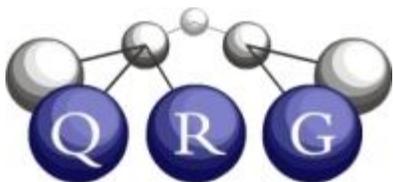
# What Can I do with the API?

- Manipulate Sketches
- (list-open-sketches)
- (get-active-sketch)
- (set-active-sketch :sketch-id <sketch id>)
- (save-sketch-to-file :sketch-id <sketch id>)
- (close-sketch :sketch-id <sketch id>)
- (open-sketch-from-file :filepath <full path to file (string)>)
- (create-new-sketch)
- (name-of-sketch :sketch-id <sketch id>)
- (user-namestring-of-sketch :sketch-id <sketch id>)



# What Can I do with the API?

- You can also manipulate subsketches, Layers and Glyphs
- (list-bundles :sketch-id <sketch id>)
- (list-layers :sketch-id <sketch id> :bundle-id <bundle id>)
- (name-of-layer :sketch-id <sketch id> :layer-id <layer id>)
- (list-glyphs :sketch-id <sketch id> :layer-id <layer id>)
- (delete-glyph :sketch-id <sketch id> :glyph-id <glyph id>)
- (ask :sketch-id <sketch id> :query <query pattern> :num-responses <positive integer or :all>)
- These are just examples of some of the available commands



You can find the whole API in the CogSketch start menu group

