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

<table>
<thead>
<tr>
<th>System</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton’s Pen</td>
<td>Draw free-body diagrams</td>
</tr>
<tr>
<td>Kirchoff’s Pen</td>
<td>Draw resistor networks</td>
</tr>
<tr>
<td>OrganicPad, ChemPad</td>
<td>Draw 2D molecules, converts to 3D</td>
</tr>
<tr>
<td>Mechanix</td>
<td>Draw trusses, get feedback</td>
</tr>
<tr>
<td>PhysicsBook</td>
<td>Draw simple mechanics problems</td>
</tr>
<tr>
<td>LogiSketch</td>
<td>Draw logic circuits, do simulation</td>
</tr>
<tr>
<td>MathPad2</td>
<td>Draw equations, do animations</td>
</tr>
</tbody>
</table>
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 than understands sketches as people do
  - Fluent, natural interaction
  - Human-like visual and spatial reasoning
  - Conceptual reasoning about the sketch
  - Domain-general

CogSketch explores these aspects
Ideas Underlying CogSketch

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

- touchesDirectly
- hasSurfaceContact
- surfaceNormal
- Edge13
- Quad3
- Ramp
- Block

Diagram showing geometric relationships and interactions such as gravity, Block, and Ramp.
CogSketch Uses Qualitative Reasoning

“The block will slide to the right and down”
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

Mappings include
- Correspondences
- Candidate inferences
- Similarity score
CogSketch Visual Modeling

Predicts human reaction time differences in geometric analogies

Captures cross-cultural effects in visual oddity task

76th 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
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.

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.

Solving ranking problems from conceptual physics textbook

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

Scott Friedman’s Ph.D.

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

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

(nameString Object-11 "glass")
(isa Object-11 GlassStemware)
...
```
Examples of Glyphs
IT'S A STICK!

IT'S A PYRAMID!

IT'S A CHRISTMAS TREE!

IT'S A CHRISTMAS TREE IN A CEREAL BOWL!

IT'S A CHRISTMAS TREE IN A CEREAL BOWL NEXT TO A SNAKE!

YOU DID SAY IT WASN'T A STICK, RIGHT?

AND YOU WONDER WHY I DON'T LIKE PLAYING THIS GAME WITH COMPANY.

WHAT'S WITH THESE "B," "O," "A" AND "T" SYMBOLS? ARE THEY FICTIONARY SHORTHAND?
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. Worksheet authoring will be discussed later. Perceptual Sketchpad and Design Coach 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

Ink that doesn’t belong to any glyph yet is shown in the thumbnail pane in grey

Finish button
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

Copy glass
Paste
- Merge Ink into glass
- Merge Ink into Object-20
- Change Color of Selected Ink
- Show Knowledge for glass
- Edit
- Properties for glass
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

1. Select the glyph to label using the Glyph lasso

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

3. Choose concept
Choosing a Concept

1. Start typing to see candidates

2. Comments help indicate which choice might be best

The collection of instances of Stemware made primarily of glass.
Choosing a Concept, continued

- Glass with water inside

Available Concepts:
- Water

Ink Properties:
- Color: blue
- Style: solid
- Width: 5 px
Relation Glyphs

• Indicate relationships between two things in the sketch
• Always drawn as arrows, as per concept maps
Labeling a Relation Glyph

A predicate that relates Social Beings 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 property, for example, a glass.

Bender property glass.
Annotation Glyphs

- Represent information about a glyph that would be hard to express visually
- Annotation glyph provides
  - Visual indicator in the sketch
  - Non-visual information
Labeling an Annotation Glyph
Types of Annotations

- AccelerationArrow
- AngleIndicator
- AngularMomentumArrow
- AppliedForceArrow
- AssumedForceArrow
- AxisAnnotation
- CarbonMassIndicator
- CCWRotationDirectionArrow
- CWRotationDirectionArrow
- direction of movement
- DirectionAnnotation
- displacement
- force arrow
- FrictionalForceArrow
- GravitationalForceArrow
- HorizontalAxisAnnotation
- HorizontalDistanceIndicator
- InitialVelocityArrow
- LengthIndicator
- MassIndicator
- MassTransferPerTimeIndicator
- MomentumArrow
- NetForceArrow
- NormalForceArrow
- PathIndicator
- PossibleSolutionIndicator
- QualitativeMechanicsAnnotation
- rotational force arrow
- rotational origin
- SketchAnnotation
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

- Each sketch must have at least one subsketch, and each subsketch must have at least one layer.
- Multiple layers in the same subsketch may be visible at the same time, like in drawing programs or acetate sheets over paper.
- 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

Use the New Layer button

Normal Layer is for sketching
Controlling Visibility
Adding a Bitmap Layer

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

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

- Describing alternatives

- Describing something from multiple perspectives
Adding a Subsketch

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

The *metalayer* lets you manipulate subsketches.
The Metalayer

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

Default = a static configuration
Conceptually Labeling a Subsketch

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

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

Initial State ➔ Falling Glass

Properties

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
A Comic Graph

Initial State → Falling glass → postEvents → Spilled Water → postEvents → Broken Glass
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 → 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)

SKETCHY (Pisan, 1994)

CLOCK Project: Example

Input: Annotated diagram

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

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

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

Examples from TU Berlin corpus (Eitz et al. 2012; 20,000 sketches spanning 250 concepts)
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
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)
Object name or glyph name
- If unbound, will query all glyphs in subsketch
Angle threshold modifier
- Adjusts flexibility in identifying straight or right angles
- Range: [-1, 2]
Query Window

Size threshold modifier
- Adjusts flexibility for calling things “same size”
- Range: [-1, 2]
Query Window

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

Gets bound to the set of edge-cycle names
Query Window

Gets bound to the number of facts generated
Query Window

Query in the Subsketch context
Query Window

(Query / WM Fact Edit)

(edgeCycleRepresentationsFor Object-7 0 0
  action = ask
  context = BCase-3662670010; facts = all, env, infer

Answers:

in BCase-3662670010:

  ? A X (edgeCycleRepresentationsFor Object-7 0 0
  (EdgeCycleFactsMtfFn 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-61105 (PerimeterEdgeCycleFn ECO-2851960)) 271)

Refresh the object pane

Return to Query/Edit Page
Query Window

Click to view facts about a perceptual entity
Query Window

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

Query in the edge-cycle microtheory.
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

(\text{edgesColinear Edge-104462 Edge-104457})

(\text{edgesParallel Edge-104455 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

Corner Attributes
- Convex, Concave
- Right, Acute, Obtuse
Edge-Level Representations

**Edge Attributes**
- Straight, Curved, Curve details: Concave/Convex, Major/Minor/Semicircle arcs
- 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, Λ, 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, Λ, L)

Positional Relations
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, Λ, L)

Positional Relations

Containment

(objectContains EdgeCycle-61107 ECO-2851959)
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

(ist-Information (EdgeCycleFactsMtFn Rabbit)
 (medialAxisRepresentationsForEdgeCycle
  (PerimeterEdgeCycleFn ECO-3735956) ?angle-mod

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

Gets bound to the list of MAT edge names

Gets bound to the number of facts

Gets bound to a microtheory containing the facts
Querying for MAT Representations

(ist-Information (EdgeCycleFactsMtFn Rabbit)
  (medialAxisRepresentationsForEdgeCycle
   (PerimeterEdgeCycleFn ECO-3735956) ?angle-mod

(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 \((R' <|\approx| 0)\)
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 ($R'' > /\approx / < 0$)
- Obtuse/Right/Acute ($R' > /\approx / < -1$)
- Source/Sink edge
- Relative thickness

Connection Radius Relations
- Weakly/Strongly directed connection
- Source/Sink connection

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

(stronglyDirectedConnection (directedConnection MedialAxisEdge-106689 MedialAxisEdge-106699))
MAT Representations

Standard Edge Attributes
- Edge
  - Directed
  - Connected
  - Obtuse/Right/Acute
- 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

\[
\text{(hasHole EdgeCycle-Texture-3848276} \\
\text{(TextureHoleFn EdgeCycle-Texture-3848276 0)})
\]

\[
\text{(hasPerimeterEdgeCycle EdgeCycle-Texture-3848276} \\
\text{(PerimeterEdgeCycleFn ECO-3847663)})
\]

\[
\text{(isa EdgeCycle-Texture-3848276 PerceptualTexture-CycleSize)}
\]

~2500 facts → ~250 facts

~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

Access analogy browser
The Analogy Browser

Browse the match
- Similarity score
- Correspondences
- Candidate inferences

Visualize the correspondences
The Analogy Display
Questions?
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

- Grouping
- Voronoi diagrams
- Positional relations
- Qualitative Topology
- Shape decomposition
- Mental Rotation
Hierarchical Representations in Psychology

Images can be represented at different levels in a spatial hierarchy

Spatial reasoning requires identifying the appropriate level

Hierarchical Representations in Psychology

Images can be represented at different levels in a spatial hierarchy

Spatial reasoning requires identifying the appropriate level

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

Jee et al. 2009

Ainsworth et al. 2011
Sketching for Learning in Science

Powerful tool for Learning
- Increases engagement
- Easier spatial reasoning
- Multi-modal reasoning

Practical Challenges
- Student resistance to sketching
- Delayed feedback (if any!) on sketches
- Too time-consuming to grade
Benefits of Intelligent Tutoring Systems

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

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

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Many STEM fields require learning <strong>spatial layouts and terminology</strong>, but students do not get timely feedback on sketching exercises.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idea</td>
<td>Develop software to coach students on sketching exercises</td>
</tr>
<tr>
<td>Domain</td>
<td>Potentially any</td>
</tr>
<tr>
<td>People</td>
<td>Maria Chang, Jeff Usher</td>
</tr>
</tbody>
</table>

### Design Coach

<table>
<thead>
<tr>
<th>Problem</th>
<th>Students have trouble using <strong>sketches to communicate their ideas</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idea</td>
<td>Develop software to coach students on their design sketches and explanations</td>
</tr>
<tr>
<td>Domain</td>
<td>Engineering Design</td>
</tr>
<tr>
<td>People</td>
<td>Jon Wetzel</td>
</tr>
</tbody>
</table>
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

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 – Draw, Finish Glyph

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
Worksheet Example – Request Feedback
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

![Image of a heart diagram with layers and feedback options]

- Layer 1: 'human-heart'
- Meta-Layer
- Current Subsketch: 'Workspace'

Your Progress:
- All Glyphs have Labels: ✔
- Required Glyphs: ✔
- Atria: ✔
- Blood Flow Relations: ✔
- Ventricles: ✔
- Overall Sketch: ✔

Updates the feedback and progress meters.
Worksheet Example – Drill Down

- 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
Worksheet Example – Blood flow arrows
Worksheet Example – Completed
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

Tolerance region defined by the worksheet author, not seen by student
Qualitative Spatial and Conceptual Differences

Sketch facts automatically generated by CogSketch, browsed by worksheet author

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.
Gradebook: grade reports

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

**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 house")

  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)

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)

Are there any important qualitative differences?

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

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

SME candidate inference indicates an important difference
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, Prof. Andrew Jacobson, Spring 2011
• Mandatory homework assignment: fault identification worksheets (3); 40 students
• Extra credit assignment: carbon cycle and greenhouse worksheets; 27 students

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
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 5th 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 5th graders think?

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

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 ≠ 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
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
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

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

Analogical Generalization
(McLure, Friedman, Forbus 2015; Kuehne et al 2000)
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 5th 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.

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.

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?”
4. Grading Rubrics

- Multiple grading criteria available (scroll down for more)
- 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
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

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.”
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
  • 5th 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
Analogue 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

<table>
<thead>
<tr>
<th>Base: Solution Sketch</th>
<th>Target: Student Sketch</th>
</tr>
</thead>
<tbody>
<tr>
<td>right-atrium</td>
<td>right-atrium</td>
</tr>
<tr>
<td>left-atrium</td>
<td>left-atrium</td>
</tr>
<tr>
<td>right-ventricle</td>
<td>right-ventricle</td>
</tr>
<tr>
<td>left-ventricle</td>
<td>left-ventricle</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Base: Solution Sketch</th>
<th>Target: Student Sketch</th>
</tr>
</thead>
<tbody>
<tr>
<td>(rightOf right-atrium left-atrium)</td>
<td>(rightOf right-atrium left-atrium)</td>
</tr>
<tr>
<td>(above right-atrium right-ventricle)</td>
<td>(above right-atrium right-ventricle)</td>
</tr>
<tr>
<td>(bloodFlows left-atrium left-ventricle)</td>
<td>(bloodFlows left-ventricle left-atrium)</td>
</tr>
<tr>
<td>(bloodFlows right-atrium right-ventricle)</td>
<td>(bloodFlows right-atrium right-ventricle)</td>
</tr>
</tbody>
</table>
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**

```
CanineGenus genls Dog

Dog genls Genls

CanineGenus genls DomesticatedAnimal

Collie is the collection of all dogs of the breed Collie

Everything that is an instance of Collie is also an instance of Dog but not vice versa
```
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 Lassiel Dog)

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

Lassiel is also an *individual*.

(isa Timmy1 MaleChild)
Predicates and genlPreds

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

(owns Timmy1 Lassie1)  
(biologicalRelatives Timmy1 Lassie1)  
Predicates can also relate Collections

(animalTypeMakesSoundType Dog BarkingSound)
(disjointWith Cat Dog)
Aritity 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
   arg1Isa: SocialBeing
   arg2Isa: 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
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 Lassiel 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
Knowledge Browser Interface

Note that the sketch and each subsketch has its own microtheory.

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

- **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

connectedAtEnd

rotationallyConnectedTo

crossed

connectedAlongSurface

alignedCylinderWithin

Above-Touching

connectedTo-Rigidly

Q

R

G
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)
How Visual/Conceptual Relations are Hypothesized

• Qualitative topology used to suggest initial candidates
  - \((\text{insideInSketch } o1 \ o2)\) if \((\text{glyph } o1)\) is inside \((\text{glyph } o2)\)
  - \((\text{atOrOverlapsInSketch } o1 \ o2)\) if \((\text{glyph } o1)\) touches or overlaps \((\text{glyph } o2)\)

• Possible specializations filtered by argument type relationships

• You can choose more specialized relationship if desired

• Not an easy problem
  - Worst case: 150 possibilities for \(\text{insideInSketch}\), 204 for \(\text{atOrOverlapsInSketch}\), 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?

- (alignedCylinder:Within 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-Roofed Front axle front wheel)
- (in-Snugly Front axle front wheel)
- (in-Region 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)
- (physicalPartsSeparated 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)
- (properSpatiallySubsumes Front axle front wheel)
- (properSpatiallySubsumes-Nontangential Front axle front wheel)
- (properSpatiallySubsumes-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)
Querying from Working Memory

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

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

“ask” is faster but doesn’t use inference

“query” will use rules in WM and KB

“tell” adds an assumption into WM

“untell” can retract assumptions in WM

Can query WM, KB, or both

“ask” is faster but doesn’t use inference

“query” will use rules in WM and KB

“tell” adds an assumption into WM

“untell” can retract assumptions in WM

“query” will use rules in WM and KB

“tell” adds an assumption into WM

“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:

<table>
<thead>
<tr>
<th>?</th>
<th>(hasRCC8Relation (GlyphFn Object-43 User-Drawn-Sketch-Layer-13) (GlyphFn Object-44 User-Drawn-Sketch-Layer-13) rcc8-PO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>(inKB (ist-Information NuSketchQMRulesMt &lt;= (touchesDirectly ?obj1 ?obj2) (wmOnly (hasRCC8Relation (GlyphFn ?obj1 ?sketch-layer) (GlyphFn ?obj2 ?sketch-layer) rcc8-PO))))</td>
</tr>
</tbody>
</table>

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-PO))))) (ist-Information BCase-3467448799 (hasRCC8Relation (GlyphFn Object-43 User-Drawn-Sketch-Layer-13) (GlyphFn Object-44 User-Drawn-Sketch-Layer-13) rcc8-PO))) (ist-Information BCase-3467448799 (touchesDirectly Object-43 Object-44)) :bc-justify-clause-bc-result)

Direct Consequences:

| ? | (touchesDirectly Object-43 Object-44) true |
Exporting Knowledge to Files

- Export Sketch Knowledge

- Sketch to be Exported: Sketch-2

- In what format should the knowledge be exported? MELD (CYC)

- In what file should it be saved? sketch-facts.txt

- Fact Filter: none

- Include detailed ink descriptions? [ ]
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

- Retrieve/store facts in the KB
- Browse the entire KB
Example: Browsing

- Let’s look for other relationships involving rotation with the KB browser
rotationallyConnectedTo [type = Relation]:

comp: 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

genlPreds:

  in BaseKB: rotationallyConnectedTo
  in UniversalVocabularyMt: connectedTo

specPreds:

  in UniversalVocabularyMt: connectedByBellTo, hingedTo, screwedin
Editing the KB through the Knowledge Browser

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

“Retrieve” looks up facts

“Store” adds fact

“Forget” removes fact
Knowledge Base Query Result

Note there is no option to drill down for justifications; these are simply facts retrieved from the KB.
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)
(genls 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 is a 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”)

Q R G
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 genlPreds 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
Analogy Browser

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

Recorded SMEs:
- SME #1, version 0

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

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

Mapping view shows what was matched with what between subsketches.
CogSketch as a Module

Reminder – you can export knowledge for use in other systems

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