Wrap-up
Overview

• Seeking feedback on some plans for the near future
  – New conceptual labeling methods
  – High-level language for visual routines
  – Authoring support for education
  – Other “sweet spots” for education?
  – Building a community

• Discussion: What would you like to do with CogSketch?
Current Conceptual Labeling Schemes

**KB concept picker**
- Wide breadth
- High reasoning support
- High entry barrier

**List concept picker**
- Narrow breadth
- High reasoning support
- Low entry barrier

**Strings**
- Wide breadth
- No reasoning support
- Low entry barrier
Glyph Button Bars

- Associate domain symbols with predefined layer types
  - Use drag and drop to fill out information
- Can scale to $10^3$ glyph types
- Only makes sense if learning visual symbols is part of domain learning

1. Choose type of glyph
2. Fill out roles in complex entities filled in via drag & drop
3. Draw
4. Draw ink for the glyph
5. Finish
Plan: Explore NLU for labeling

• Use string as input to natural language system
  – Lexical lookup
  – Phrase parsing
  – Use context of sketch to help disambiguate
  – If uninterpretable, fall back to just recording string

• Resources
  – WordNet/OpenCyc links already in KB
  – Exploring VerbNet for subcat frames to import
  – Existing simplified English NLU system (EA NLU) using ResearchCyc KB contents
Modalities for entering non-spatial information

• Examples: intended behavior, purpose of design, Q/A in tutoring, …

• Simplified English NLP
  – Same infrastructure for conceptual labeling, plus discourse processing
  – Progress in language-based tutors suggests that this may be feasible for particular types of tasks

• Form-filling
  – Much less flexible, but very practical
  – Can use same word/phrase parsing as conceptual labeling
Smoother Interface Mechanics

• Annoyance: Button presses to start/stop glyph drawing
  – Improvement: Right-click for start/stop as option
  – Speech commands another option in some settings
  – Open question: How can automatic segmentation be made usable?
    • Needs to be extremely reliable
    • Needs robust error recovery
    • Need to allow user intervention if necessary
Continue extending CogSketch into broad-scale model of human visual-spatial processing

- Accurate simulation of human performance
  - Evans, RPM, Visual Oddity
  - Expand: Sorby, Vandenberg, Paper Folding, etc.
- Explain individual & group differences via parameters and ablation
- Goal: Convergence of model as number of phenomena captured grows

![Graph showing accuracy vs. model additions and phenomena modeled.]
Visual Routines Language

- Rapid convergence of techniques used to solve various visual tasks
  - But all of the simulations are written in Lisp code, driving CogSketch internal operations
- Possible approach: Define high-level declarative language for writing visual routines
  - Constrained to psychologically plausible operations
  - Support uploading of new routines by CogSketch users
  - May provide a simpler way to program CogSketch than the API
Other “sweet spots” in education?

• Working hypothesis: Sketch-based educational software could have great benefits for education
• Worksheets: Simple, practical, low entry barrier
• Design Buddy: Complex, but could raise the bar for intelligent tutoring systems
• Where else should we be looking?
Building a Community

• Add “phone home” facility for gathering data from willing users
  – Identities scrubbed for privacy reasons
• Provide on-line archive for researchers to access submitted sketches
• Build distribution site for worksheets
  – Goal: Create an “ecology” of worksheet users
Discussion

• You’ve now seen the current state of CogSketch
• What might you be interested in doing with it?
• How might we extend it to help you do that?