Sketch Recognition for Course of Action Diagrams

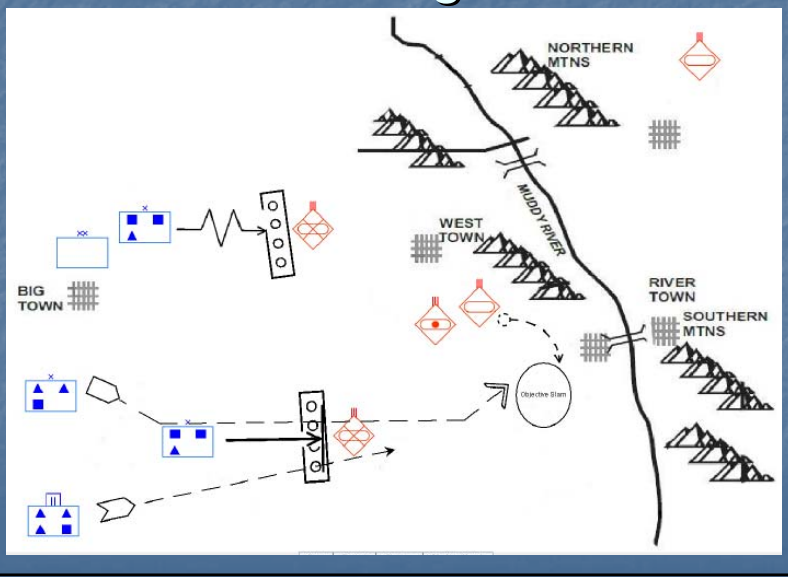
Kevin Stolt

Presented by
Freddie Santiago

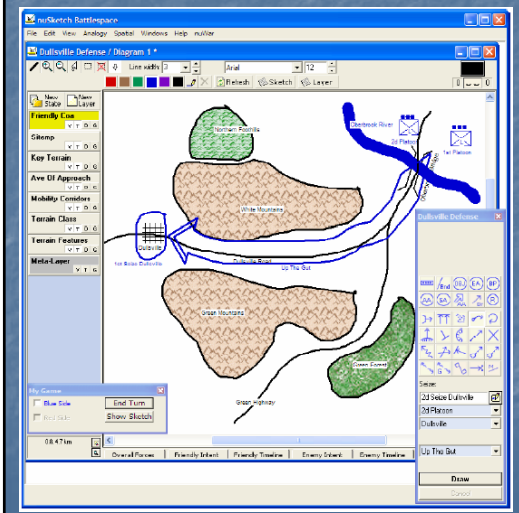
Introduction

- A course of action(COA) diagram is created by military planners when they are formulating a battle plan.
- COA diagrams consist of symbols representing military units and the actions they perform.
- Many COA diagrams today are created using pen and paper, overlay maps and grease pencils, post-its, and pushpins.
- Many computerized attempts at speeding up the process have been rejected because of the awkwardness of mice and menus versus hand sketching.

COA Diagram



nuSketch Battlespace



- Relies on glyph bar to create units, and drop down menus for unit properties, and actions.
- Avoids recognition problems
- Creates complex and inefficient interface.

Visual Classification Motivation

- Shimon Ullman (Nature Neuroscience Journal) found that the features of intermediate complexity are most optimal for object classification because of specificity and relative frequency
- When a human looks at a picture or sketch, they perform object classification quickly, easily recognizing familiar object in the picture.
- Research suggests that human visual processing begins by using simple local features, then subsequently representing more complex features.

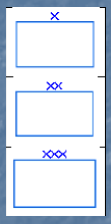
LADDER

```
(define shape Arrow
  (description "An arrow with an open head")
  (components  (Path shaft)  (Line head1)  (Line head2))
  (constraints
    (coincident shaft.p1 head1.p1)
    (coincident shaft.p1 head2.p1)
    (coincident head1.p1 head2.p1)
    (equal-length head1 head2)
    (acute-meet head1 shaft)
    (acute-meet shaft head2))
  (aliases (Point head shaft.p1) (Point tail shaft.p2))
  (display
    (original-strokes shaft) (cleaned-strokes head1 head2) (color
   red))
  (editing
    ((trigger (click_hold_drag shaft))
      (action (translate this) (set-cursor DRAG)
              (show-handle MOVE tail head)))
    ((trigger (click_hold_drag head))
      (action (rubber-band this head tail) (show-handle MOVE head)
              (set-cursor DRAG)))
    ((trigger (click_hold_drag tail))
     (action (rubber-band this tail head) (show-handle MOVE tail)
            (set-cursor DRAG)))) )
```

Course of Action Diagram

- Military Symbolic language to visually represent various aspects of military operations.
- Unit Symbol represents unit's strength, size, branch, affiliation, dimension, composition.
- Operation Symbol – tasks to be performed.

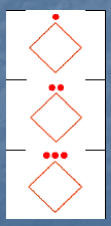
Course of Action Diagram



Brigade

Division

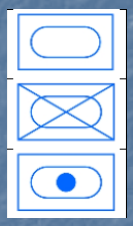
Corps



Squad

Section

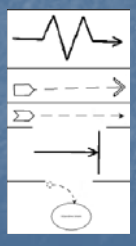
Platoon



Armor

Mechanized Infantry

Self-Propelled Howitzer



Fix

Follow and Assume

Follow and Support

Penetrate

Seize

System Functionality Sketch

Shape creation
- Can draw strokes in any order and at any scale.
- Strokes replaced by an image

Deletion
- Scribble

Moving
- Touch pen on unit for half a second, cursor switches
- Drag the pen to new location

System Functionality Multimodal

Used to communicate information which may be too difficult or impossible to communicate through pen-based input.

Naming
- Talk button
- “This is <symbol name>”, while clicking once on the symbol.
- Talk button

Copy/Move
- Talk
- “Copy this unit here”, while clicking once for the unit, and once for the destination
- Talk

System Components

Three main system components:

- Sketch Recognizer – recognizes COA sketched shapes
- COA Domain Handler – ensures only valid sketch combinations are recognized
- COA Multimodal Recognizer – combines pen and speech input

Sketch Recognizer

Three components:

- Primitive Recognizer
- Intermediate Feature Recognizer (IFR)
- Domain (LADDER) Recognizer

System Components

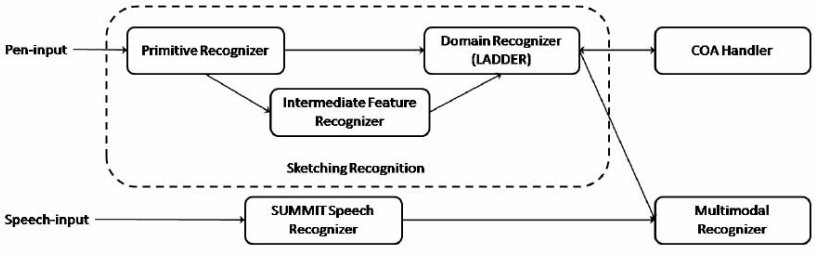


Figure 4-1: System Components

Primitive Feature Recognizer

- Classifies single strokes into primitive shapes.
  Ellipse, line, point, polyline, scribble, etc.
- Only keeps important reference points (i.e. line endpoints)
- A stroke can be classified ad multiple primitives. If so, each recognized primitive is passed on
- Each classified shape has attribution which is also passed on (i.e. ellipse height, width, center)
- Each stroke has a UID which is used so that only one makes it through the domain recognizer

Primitive Feature Recognizer

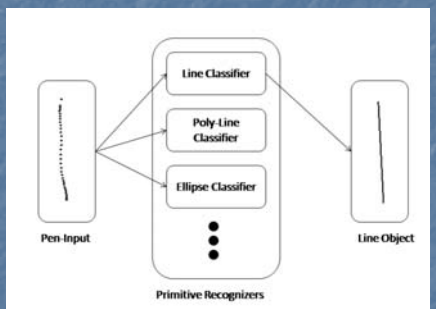


Figure 4-2: Pen-Input Data Classified as Line Primitive Type

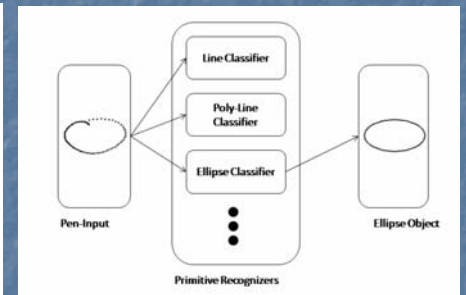


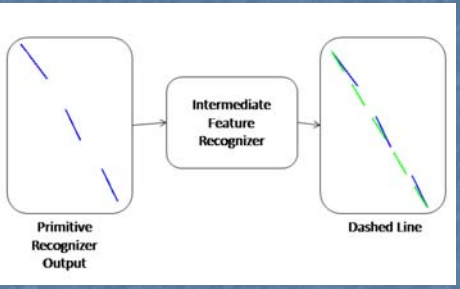
Figure 4-3: Pen-Input Data Classified as Ellipse Primitive Type

Intermediate Feature Recognizer

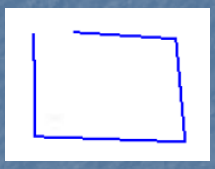
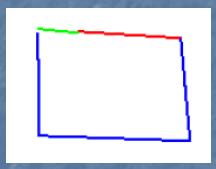
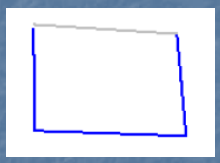
- Recognizes shapes of intermediate complexity, following on results from visual classification
- Analyzes primitive recognizer objects, and combines them into shapes of intermediate complexity.
- Passes data to the Domain Recognizer, on top of the data from the Primitive Recognizer.

Intermediate Feature Recognizer

- Able to recognize Dashed lines
  - Look to see if they are collinear, same slope
- Dashed chains – Combines two dashed-lines if they are close enough



Intermediate Feature Recognizer Error Correction

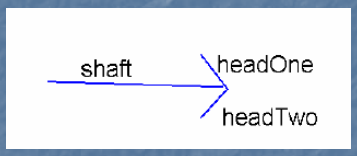




Domain Recognizer

- Uses the LADDER shape definition it recognizes the symbols
- The collection of shapes used to recognize the symbol are placed in the Visible Shape Collection (VSC)
- When new shape added, the DR attempts to combine it with other shapes in the VSC to produce a more complex shape. If it can, all components are removed from the VSC, and the new complex shape is added.

Domain Recognizer

Arrow Example



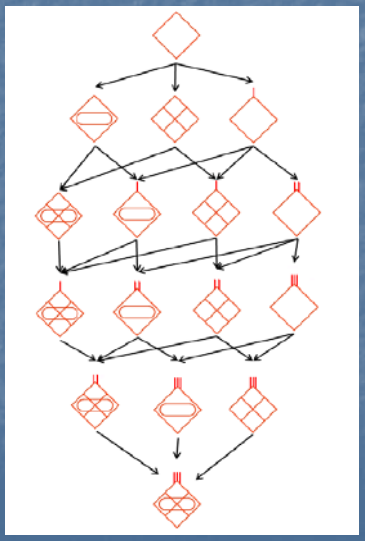

```
(define shape Arrow
    (components
        (Line shaft)
        (Line headOne)
        (Line headTwo))
    (constraints
        (coincident shaft.p1 headOne.p1)
        (coincident shaft.p1 headTwo.p1)
        (coincident headOne.p1 headTwo.p1)
        (equal-length headOne headTwo)
        (acute-meet headOne shaft)
        (acute-meet shaft headTwo))
    ...
    )
```

LADDER Problem

- LADDER works fine when there is a single way of defining a shape
- Has a hard time defining shapes that could have variable construction.

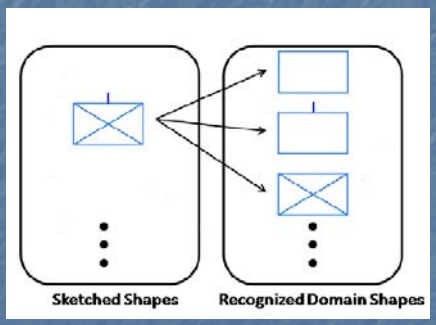
LADDER Problem

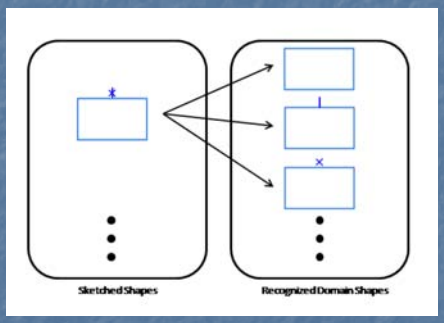
- 20 unique orders to draw the shape
- Defining a shape based on a change from a previous shape would restrict the system.
- Solve this by recognizing things on the context of a frame



COA Domain Handler

Recognizing based on a frame context causes problems



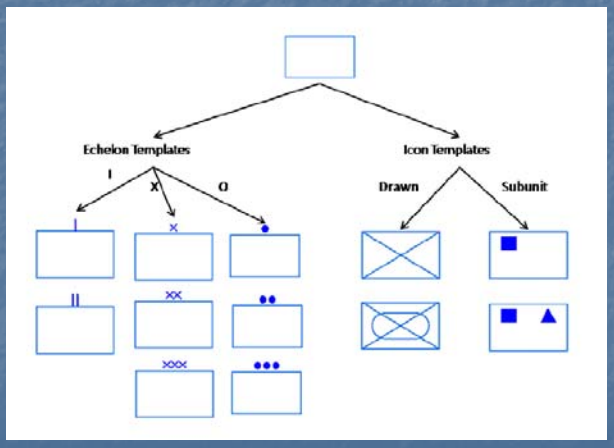


COA Domain Handler

- Fixed by using frame templates for each of the shapes.
- Instead of recognizing shapes in the context of a frame, they are recognized in the context of a frame template.
- Once a modifier has been added, the template changes, turning off any paths which would be incorrect.

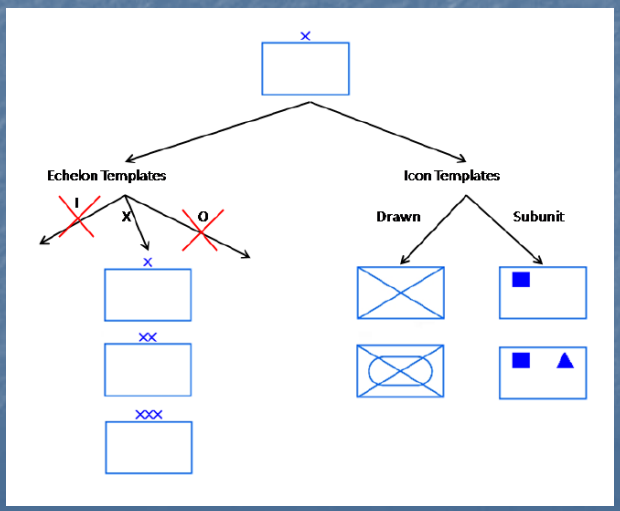
COA Domain Handler

Frame Template



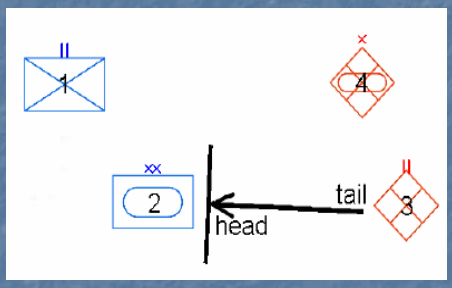
COA Domain Handler

Modified Frame Template



COA Domain Handler

- DH also interprets the scene.
  - internal representation of each symbol
  - updates information about them whenever changes occur.
  - Aggressor/Defender



Multimodal Recognizer

Speech Input

- Speech is broken down into small units of sound called phonemes
- A sequence of matches is used to produce and n-best list of candidate sentences accompanied by a score.

Combining Speech and Pen

- A type of command is matched with an expected number of inputs from the pen.
- If there is a match, then the action is performed, Otherwise, nothing is done.

Multimodal Recognizer

Copy Command

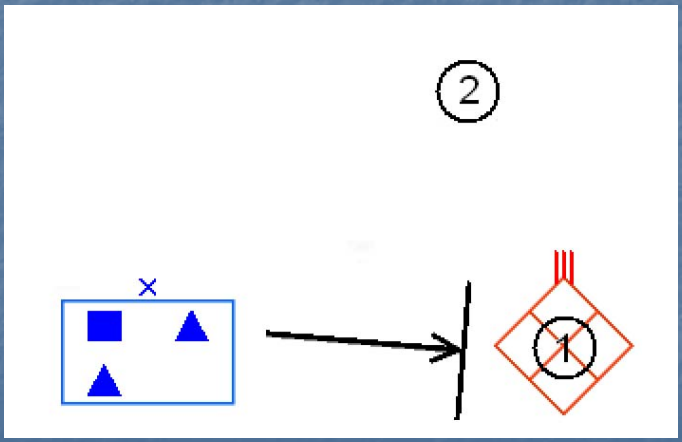
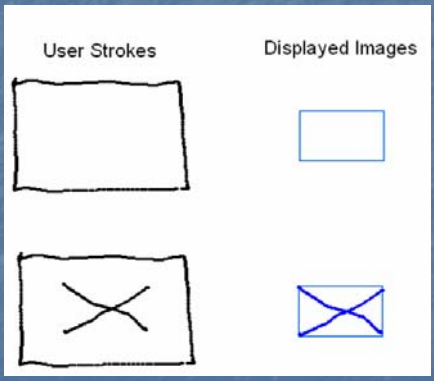
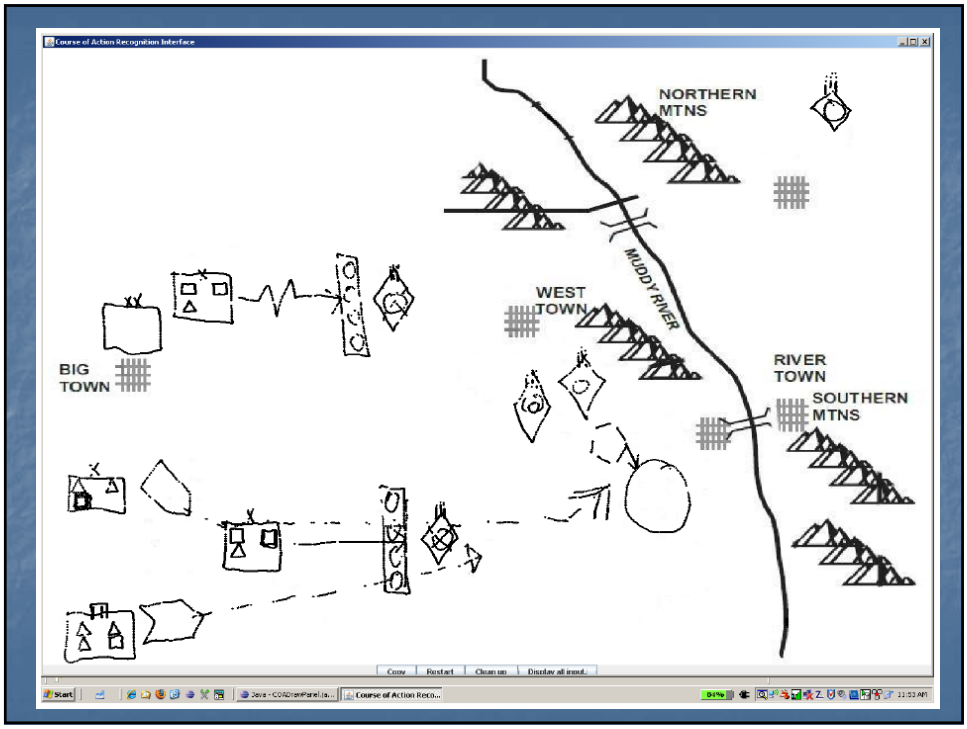
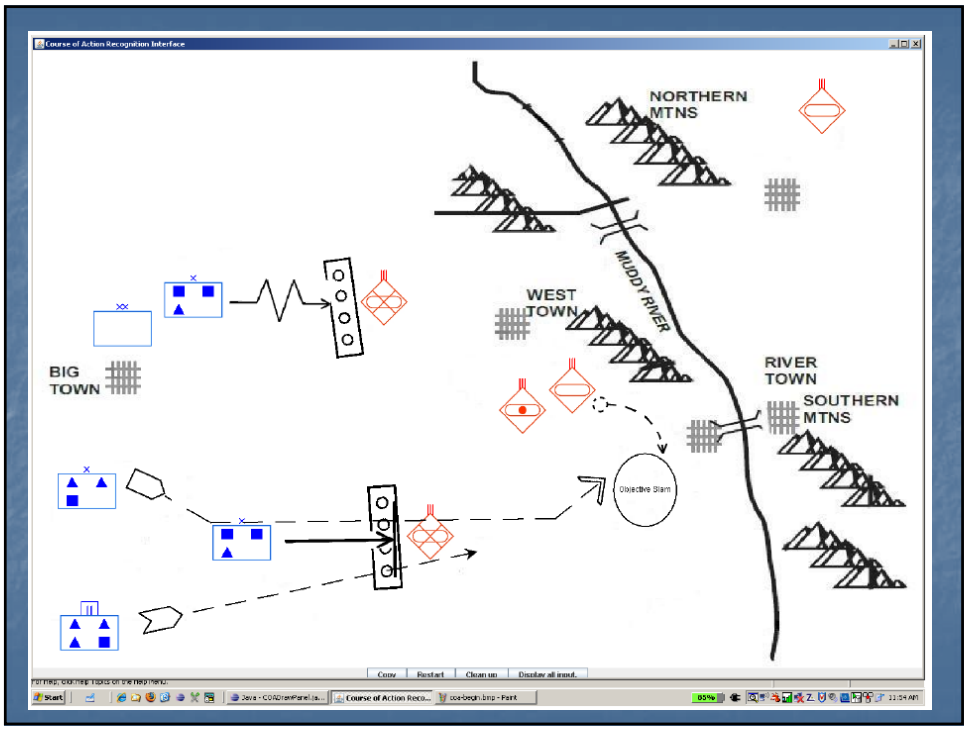


Image Replacement Problem

- Image differs from underlying shape data.
- Attempting to add modifiers to the shape may not be correct.
- Solution: Modify the strokes to match up with the picture.




Course of Action Recognition Interface
NORTHERN MTNS
MUDDY RIVER
WEST TOWN
RIVER TOWN
SOUTHERN MTNS
BIG TOWN
Copy
Restart
Clean up
Display all input.


Course of Action Recognition Interface
NORTHERN MTNS
MUDDY RIVER
WEST TOWN
RIVER TOWN
SOUTHERN MTNS
BIG TOWN
Objective Slam
Copy
Restart
Clean up
Display all input.

Conclusion

- Seems like a fairly straightforward system, but where are the performance numbers?
  - User tests?
  - Recognition percentages?
  - Learning curve of spoken commands?
- How does expensive in “online” recognition of this kind, since every stroke has to be compare to existing features in the VSC during the IFR and COA domain recognizer?

Improvements

- Add Zoom functionality so that macro and micro echelon battle strategies are displayed as you zoom