Research in
Integrated Development Environments
(IDEs)

presented by
Gene Fisher

What Is an IDE?

- Integrated collection of tools.

- Basic components are:

  o source code editor

  o compiler/interpreter

  o debugger

Advanced IDE Features

- multi-language support

- code browsing

- graphical views

- testing

- integrated version control

Earliest IDEs

- Lisp environments of the 1970s.

- First terminal-based, then GUI.

The Lisp IDE Experience, Then and Now

```
> (defun avg (l)
     (/ (sum-list l) (length l)))
AVG


> (defun sum-list (l)
     (if (null l) 0
         (+ (car l) (sum-list (cdr l)))))
SUM-LIST


> (avg '(1.0 2 3 4 5 6 7 8))
4.5
```

The Lisp IDE Experience, cont'd

```
> (bye)
Bye.
```

Research Heyday

- Mid-1980s.

- Pioneering environments like

  - PECAN -- the original Eclipse

  - BALSA -- algorithm animation

  - Daisy II -- mixed language environment

  - Demo -- programming by demonstration

PECAN -- The Original Eclipse

- Provided multi-window viewing and editing.

- An command-line execution window.

- A GUI debugger.

- Stack and data views.

- Syntax-directed editor.

- Incremental compilation.

Figure 8: An execution view

BALSA -- Pioneer in Algorithm Animation

- Provided a wide variety of ways to visualize program behavior.

- E.g., watch as the elements of a list are sorted.

- Visualize complex data structures dynamically.

- E.g, next page shows animation of a compiler.

procedure expression;

procedure term;

procedure factor;

procedure expression;

procedure expression;

procedure term;

```
procedure factor;
  begin
  if p[j]='(' then
    begin
    j:=j+1;
    expression;
    if p[j]=')' then j:=j+1 el
    end
  else if letter p[j] then
  if p[j]='*' then j:=j+1;
  end;
```
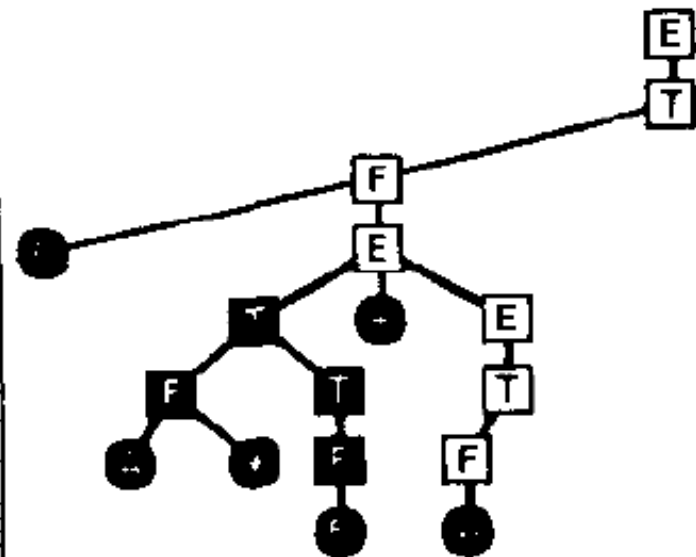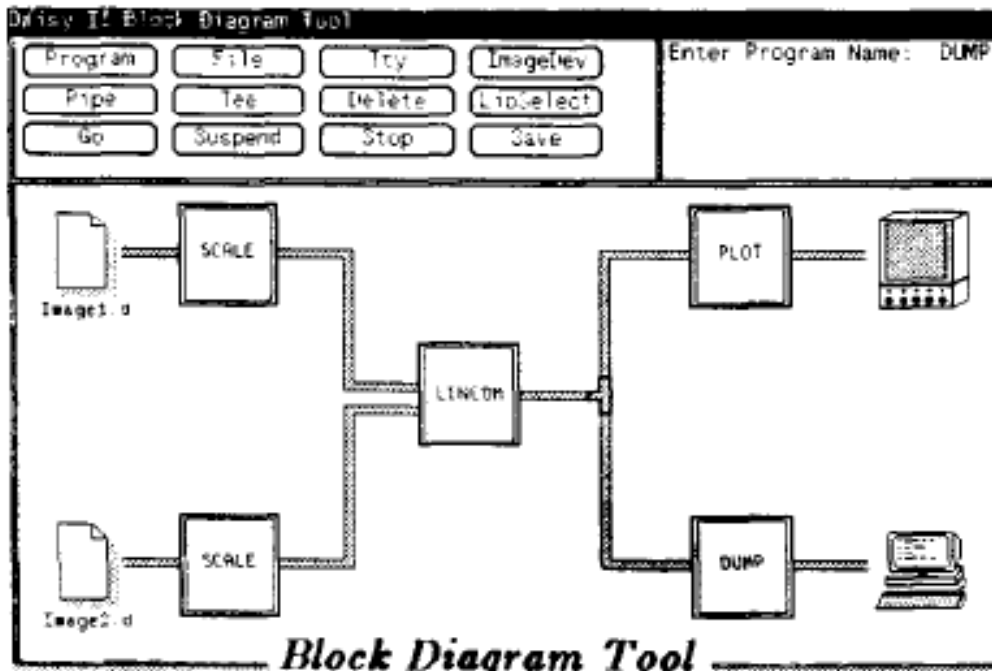
$(A*B+AC)D$

(A * F + - C) D

Daisy II -- mixed language environment

- Allowed development in a combination of differ-ent languages.

- Provided BALSA- and PECAN-like viewing features.

- Also provided visual input devices.

- And a graphical dataflow tool.

```
[ Program ]  [  File  ]  [   Tty   ]  [ ImageDev ]      Enter Program Name:   DUMP
[  Pipe   ]  [  Tee   ]  [ Delete  ]  [ LibSelect ]
[   Go    ]  [ Suspend ] [  Stop   ]  [  Save    ]
```

Block Diagram Tool

Image1.d → SCALE → LINCOM → PLOT

Image2.d → SCALE → DUMP

Daisy II Panel   Procedure LINCOM

```
PARAMETER: 'Coeff3'
TYPE    =    Real
CLASS   =    variable
DEFAULT =    42.000
VALUE   =    42.000\expr=42]
HISTORY =    [ 46.000, 50.000, 55.000, 65.000, 0.000]
HELP    =    Additive coefficient
```

ERROR WINDOW: No errors

```
[  Run   ] [ Run&Done ] [ Abort ] [ PanelHlp ] [ ProgHlp ] [ ParmHlp ]
[ AltView ] [ Inc/Dec ] [  Zoom  ] [ Default ] [ History ] [  Undo  ]
```

```
1.  Coeff1
[ 3.000
2.  Coeff2
[ 10.000
3.  Coeff3
[ 42.000 ]                        ↖
4.  Infile1
[ <== SCALEa.OutFile
5.  Infile2
[ <== SCALEb.OutFile
6.  OutFile
[ ==> TEE
```

Control Panel

Daisy II Emacs Edit   File: LINCOM.p

```
{Performs linear combination of two image files:
 OutFile = Coeff1*InFile1 + Coeff2*InFile2 + Coeff3 }
procedure LINCOM(
        Coeff1,        {Multiplicative coefficient for 1st input file}
        Coeff2,        {Multiplicative coefficient for 2nd input file}
        Coeff3         {Additive coefficient}
            : real;

        Infile1, Infile2       {Input image files}
            : ImageFile;

        OutFile                {Output image file }
            : Ir
```

Language-Sensitive Editor

Daisy II Multi Command Exec   file daisy.start

```
P> #include " ./src/lib/LINCOM.p"      {load LINCOM}

P> LINCOM;             {no parameters means invoke control panel}
```
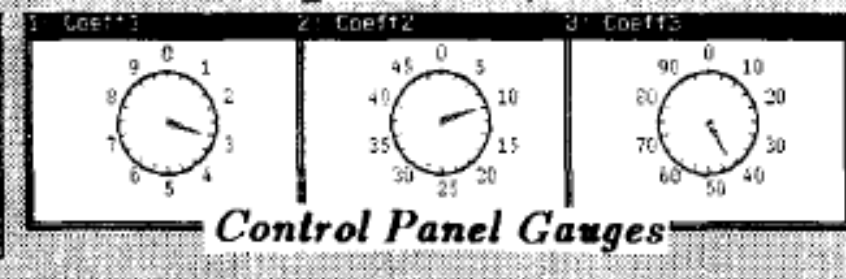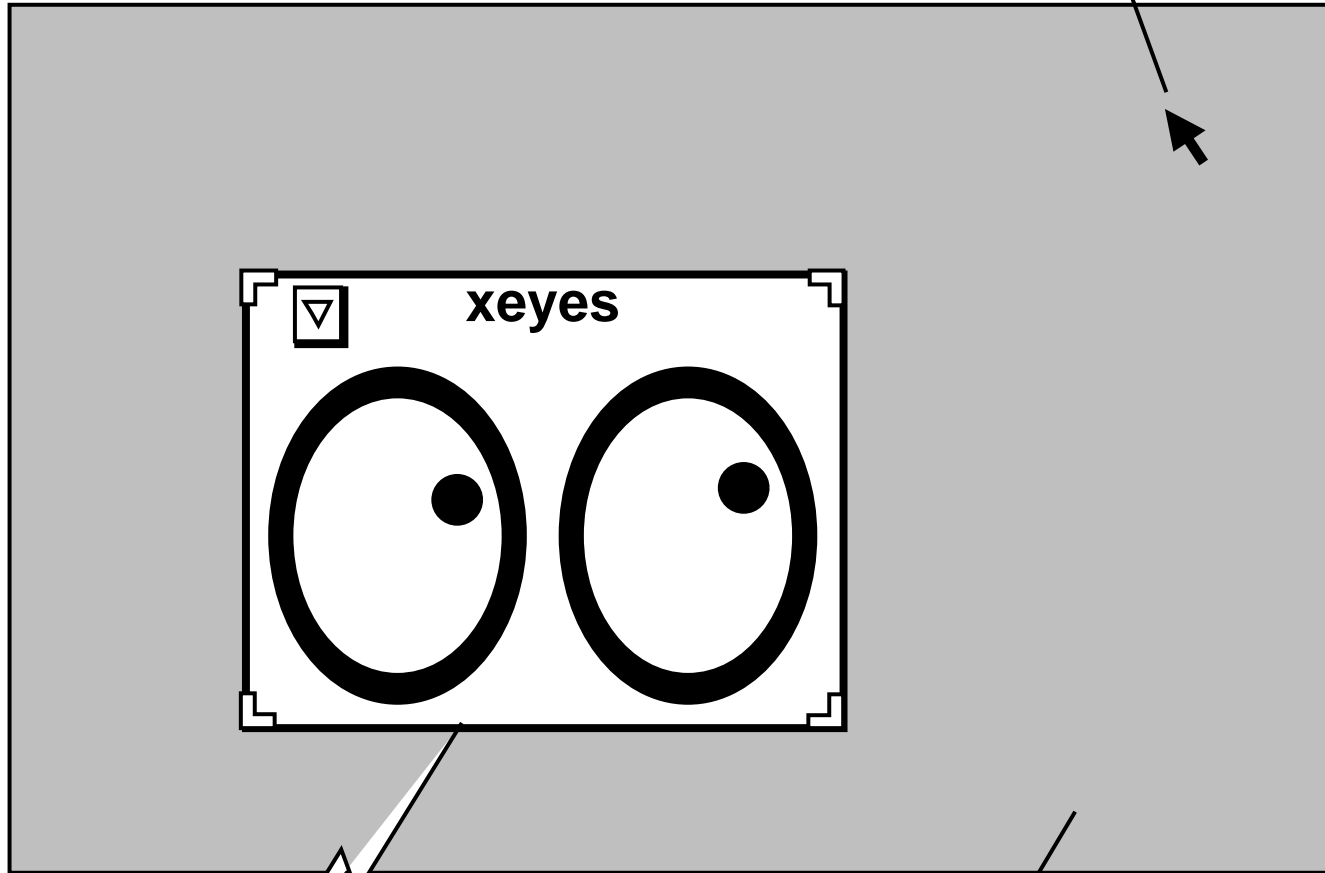
Multi-Language Command Interpreter

1: Coeff1   2: Coeff2   3: Coeff3

Control Panel Gauges
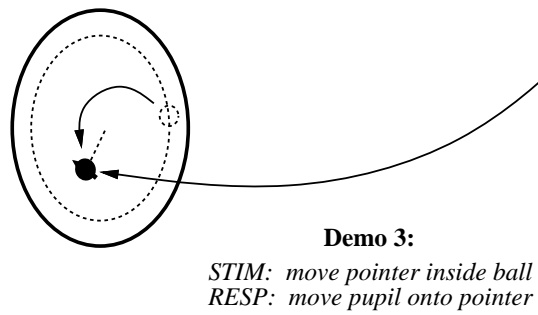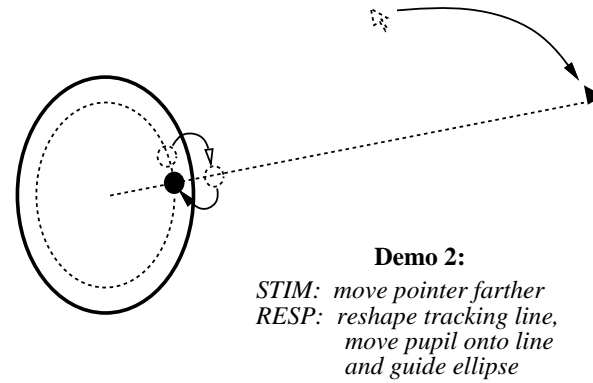
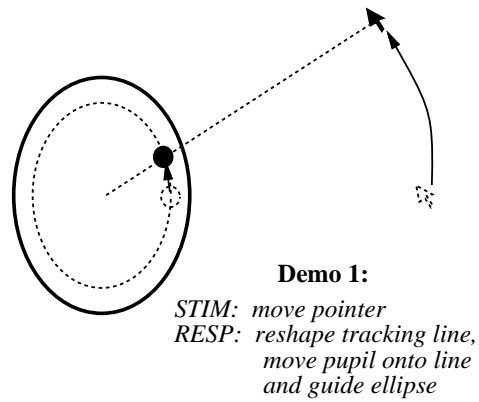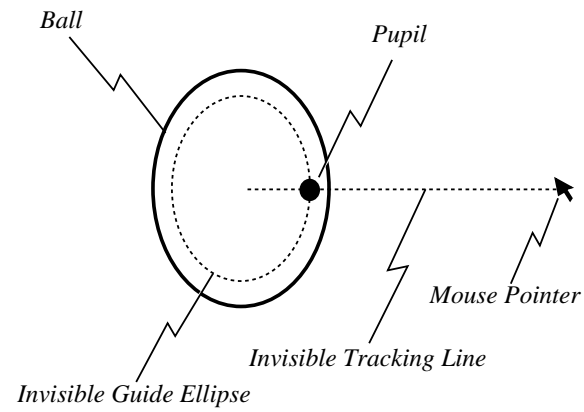# Demo -- programming by demonstration

- Allowed graphical demonstrations of GUI behavior.

- General behavior inferred from demonstrations.

- Expert system used to make inferences.

- Code generated.

mouse pointer that xeyes "watches"

xeyes

xeyes display

background screen

Ball

Pupil

Mouse Pointer

Invisible Tracking Line

Invisible Guide Ellipse

Demo 1:
STIM: move pointer
RESP: reshape tracking line,
move pupil onto line
and guide ellipse

Demo 2:
STIM: move pointer farther
RESP: reshape tracking line,
move pupil onto line
and guide ellipse

Demo 3:
STIM: move pointer inside ball
RESP: move pupil onto pointer

Demo 4:
STIM: move pointer inside ball
RESP: move pupil onto pointer

Productization

- 1990s and 2000s have seen many commercial and open source products.

- Eclipse is noteworthy.

- It has essentially nothing new compared to earlier research environments.

- Does provide a production-quality environment.

Latest Research

- Focus on expanding IDE scope.

- Areas include:

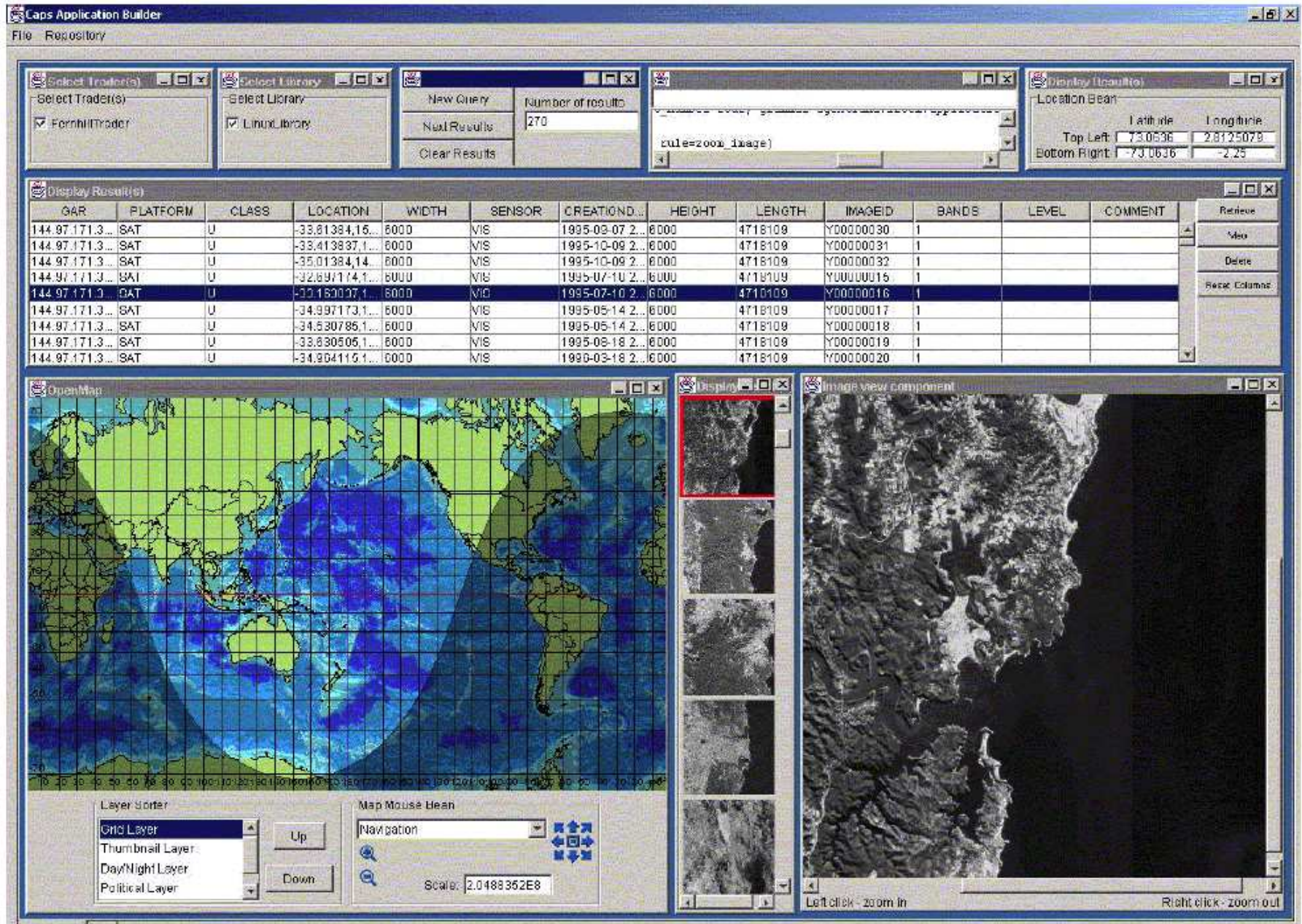 o hypermedia (Chimera)

  o collaborative work (Jazz)

  o cooperative ubiquitous computing (iRoom)

Figure 3 The iMedia application in the iRoom