

	Year	LoC	
Multics	1964	300K	
Genera 8.3	1981	1M	
Linux 1.0	1994	170K	Evans 2013
Google	2015	1B	P. Doughty-White, M. Quick 2015

usable
abstractions for secure distributed programming"

LISP (McCarthy 1958-)
Time sharing (McCarthy 1959)
Burroughs B5000 (1961)
"On-Line Man Computer Communication"
(Licklider 1962)
Sketchpad (Sutherland 1962)
IBM System/360 (1964)
Multics (1965)
oN-Line System, "Mother of all Demos"
(Engelbart 1968)
Smalltalk (Kay+ 1967-)
Unix 1.0 (Thompson 1969)
GNOSIS, KeyKOS (1972, 1975)
Lisp Machines (1970s/1980s)
Plan9 (Thompson, Pike 1992-2015)
Relational Programming
(Friedman, Byrd 2009-)
STEPS (Kay, 2012-2015)
Urbit (Yarvin, Wolfe-Pauly 2013-)
Eve (Granger 2015-2017)
Dynamicland (Victor 2018-)
Dark (Biggar 2018-)
Unison (Chiusano 2019)
wasm (2015-)



Burroughs B5000 1962 B5500 1964

Features: ([Mayer 1982](#), [Organick 1973](#))

- > Virtual memory, W^X
- > Tagged words
- > HW stack, no (accessible) registers
- > Descriptors: HW bounds checking
- > OS in ALGOL 60, no assembly
- > String instructions
- > User definable instructions
- > Dynamic reconfiguration

Descendants:

- > IBM System/38, AS/400, IBM i
- > Intel i432
- > Unisys ClearPath MCP

```
5:ESPOL/DISK= 4:SAVE=3422 OLAY=3234
8:LOOP2/TEST= 5:SAVE=698 OLAY=15
5:ALGOL/CHECKAL= 6:SAVE=3211 OLAY=4649
TOTAL MEM IN USE= 31612

TS
PBD/0092001 REMOVED
5:ALGOL/ALGOL=01 IN FOR 3:26, NEEDS 12608
5:ALGOL/COBOL=03 IN FOR 2:52, NEEDS 12608
5:ESPOL/DISK=06 IN FOR 1:42, NEEDS 10496
5:ESPOL/DISK=07 IN FOR 1:41, NEEDS 10496
-FLAG BIT ESPOL/DISK= 4, S= 3, A= 197
CDF REL ESPCLD 1:ESPOL/DISK= 4
PRNPBT/DISK= 3 EOJ 0752
DECK#0048 REMOVED
0:PRNPBT/DISK= 2 BOJ 0752
ESPOL/DISK= 4 DS-ED 0752
5:ESPOL/DISK= 3 BOJ 0752
PBD/0097001 REMOVED
5:ESPOL/DISK=02 SCHEDULED 0752
CDB IN CDCOOL 1:ESPOL/DISK= 3
PRNPBT/DISK= 2 EOJ 0752
PBD0098 OUT COOL LISTING:ESPOL/DISK= 3
0:PRNPBT/DISK= 2 BOJ 0752
PATCH/MERGE= 1 EOJ 0752
-FLAG BIT ESPOL/DISK= 3, S= 3, A= 200
CDF REL CDCOOL 1:ESPOL/DISK= 3
DECK#0049 REMOVED
0:PRNPBT/DISK= 1 BOJ 0752
5:ALGOL/ALGOL= 4 BOJ 0752
ESPOL/DISK= 3 DS-ED 0752
PBD/0098001 REMOVED
5:ALGOL/ESPOL=01 SCHEDULED 0753
PRNPBT/DISK= 1 EOJ 0753
```

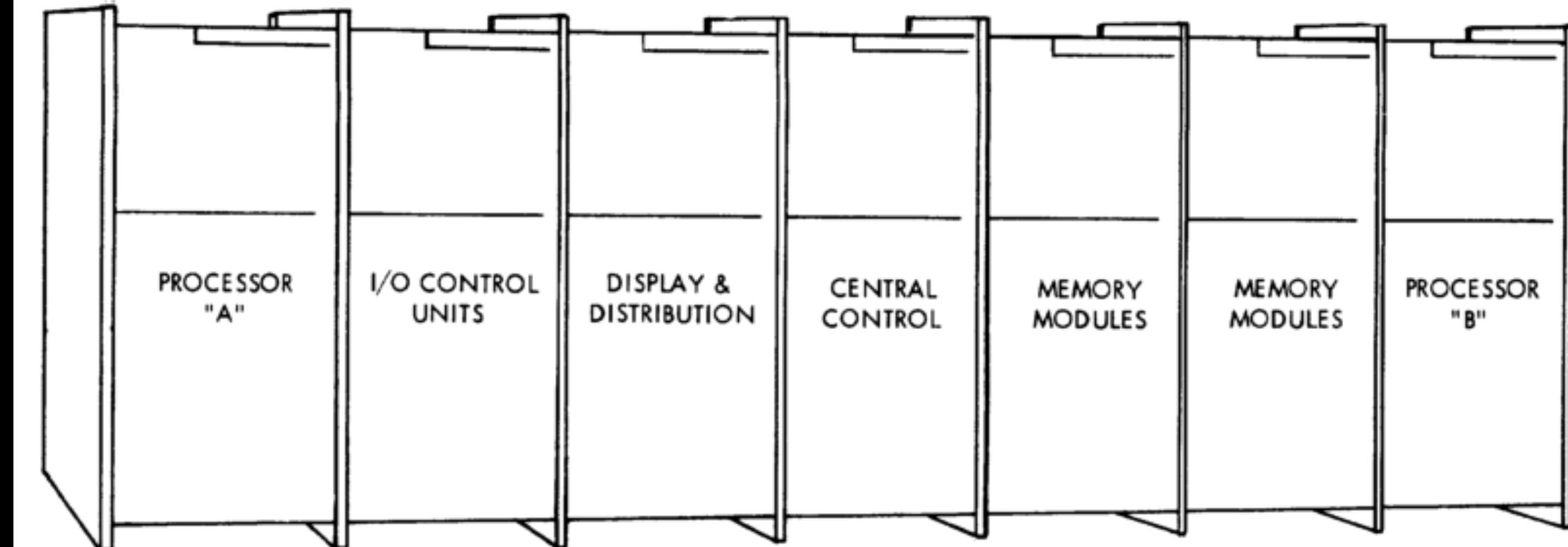
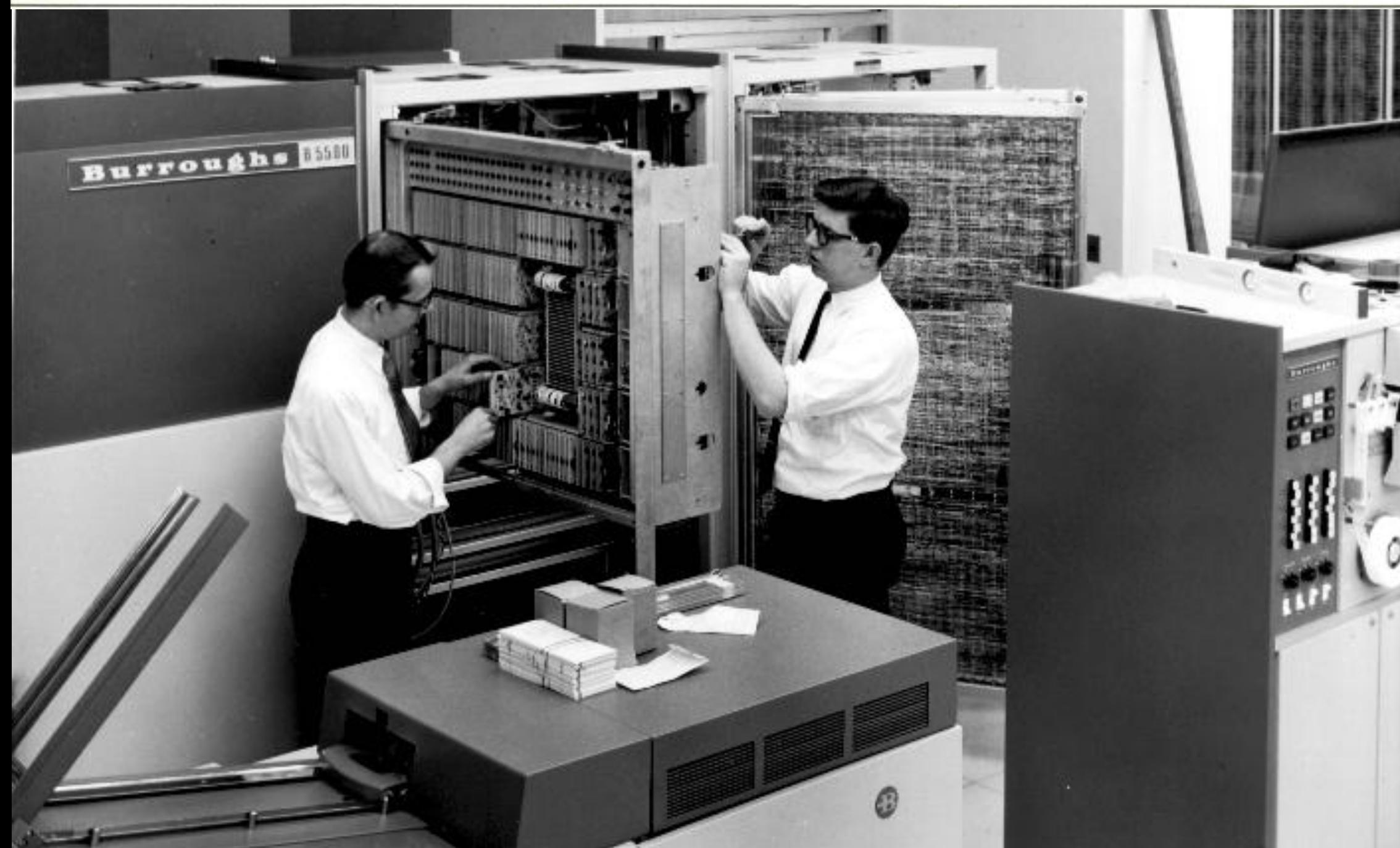


Figure 1-1. Major System Cabinet Configuration



Multics 1964-2000 MIT, GE, Bell Labs, Honeywell

Features:

- > Hierarchical fs: symlinks, quotas, removable devices, ACLs ([Daley, Neumann 1956](#))
- > Virtual memory: segments, paging, "W^X", "ASLR", virtual cpu, SLM ([Daley, Dennis, Van Horn 1958](#))
- > "Dynamic linking": call by name, demand paging, condition system ([Corbató, Vyssotsky 1965](#))
- > Virtual deadline scheduler: load control groups, non-/real time response specs ([Mullen 1976](#))
- > Live instrumentation ([Gintell, Saltzer 1969](#))
- > Dynamic reconfiguration "computer utility" ("cloud"): CPU, memory, drives, peripherals ([Schell 1984](#))
- > First commercial RDBMS ([Weeldreyer, Friesen 1978](#))

Security:

- > PL/1 prefix-length strings
- > W^X segmentation
- > random uid addressing
- > Upwards stack growth
- > Tagged inter-procedure pointers (uid names)
- > Coded null pointers (-n)

But:

- > "Complex", expensive processes
- > Ambient authority / No POLA
- > Unics



Fernando Corbató

```
Load = 6.0 out of 90.0 units: users = 6, 09/29/19 1443.5 est Sun
l Zanarotti.StudentTZ
Password:
You are protected from preemption until 14:43.
Zanarotti.StudentTZ logged in 09/29/19 1443.6 est Sun from ASCII terminal
\ce".

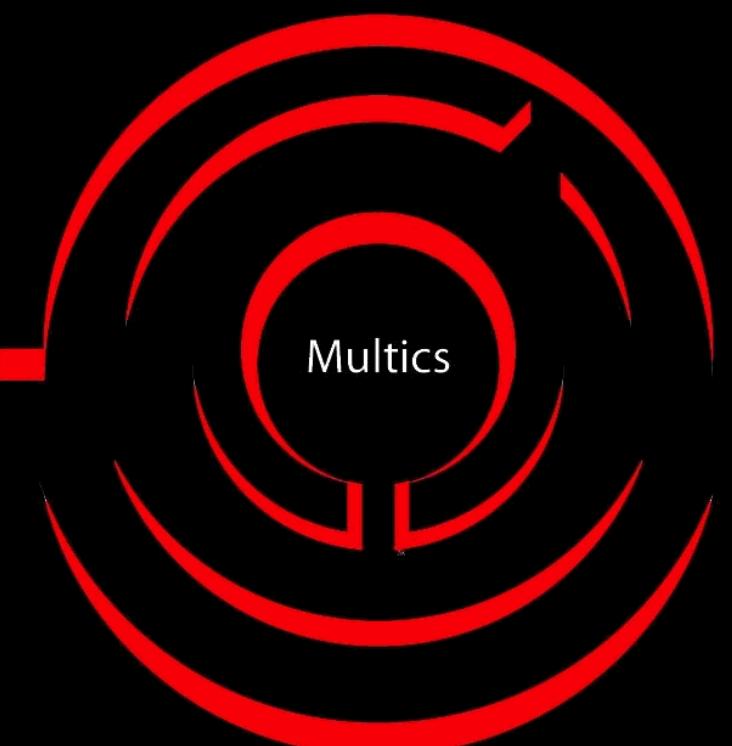
New messages in message_of_the_day:

Welcome to the Multics System.

print_motd: Created >user_dir_dir>StudentTZ>Zanarotti>Zanarotti.value.
r 14:43 0.605 20

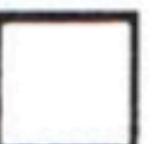
who

Multics MR12.6f, load 7.0/90.0; 7 users, 2 int
Absentee users 0/3
Rochlis.SIPBADMIN
Zanarotti.StudentTZ
r 14:43 0.052 0
```



Smalltalk Kay, Ingalls, Goldberg, Merry, Kaepler, Wallace, Deutsch (1967-1980)

```

box new named "joe"
box: joe
  

joe turn 30!
ok
  

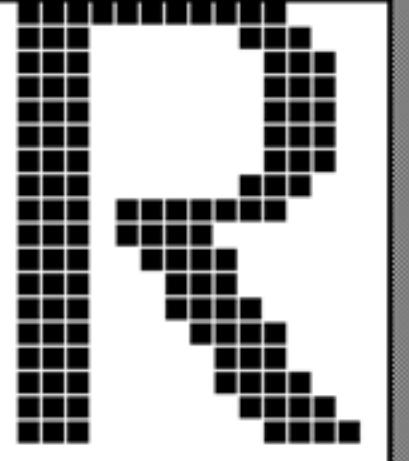
joe grow -15!
ok
  

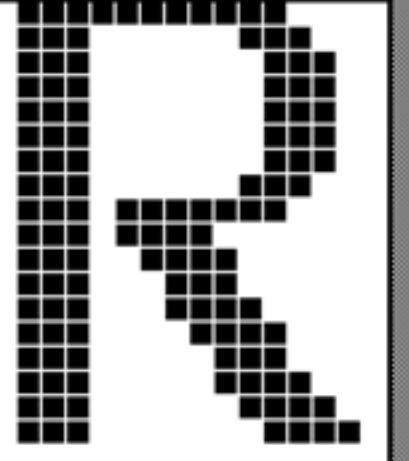
joe erase!
ok
  

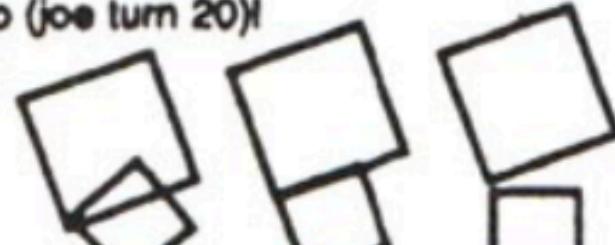
joe show!
ok
  

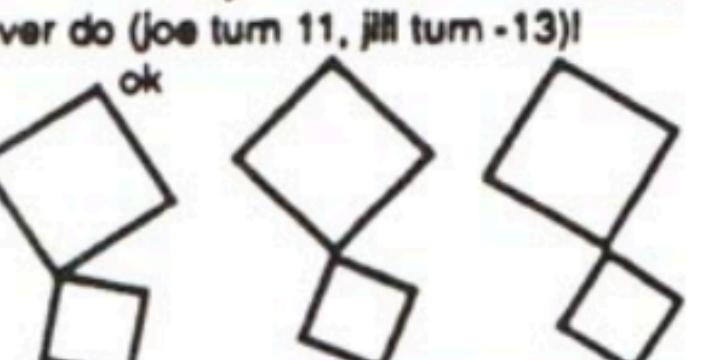
box new named "jill"
box: jill
  

jill turn -20!
ok
  

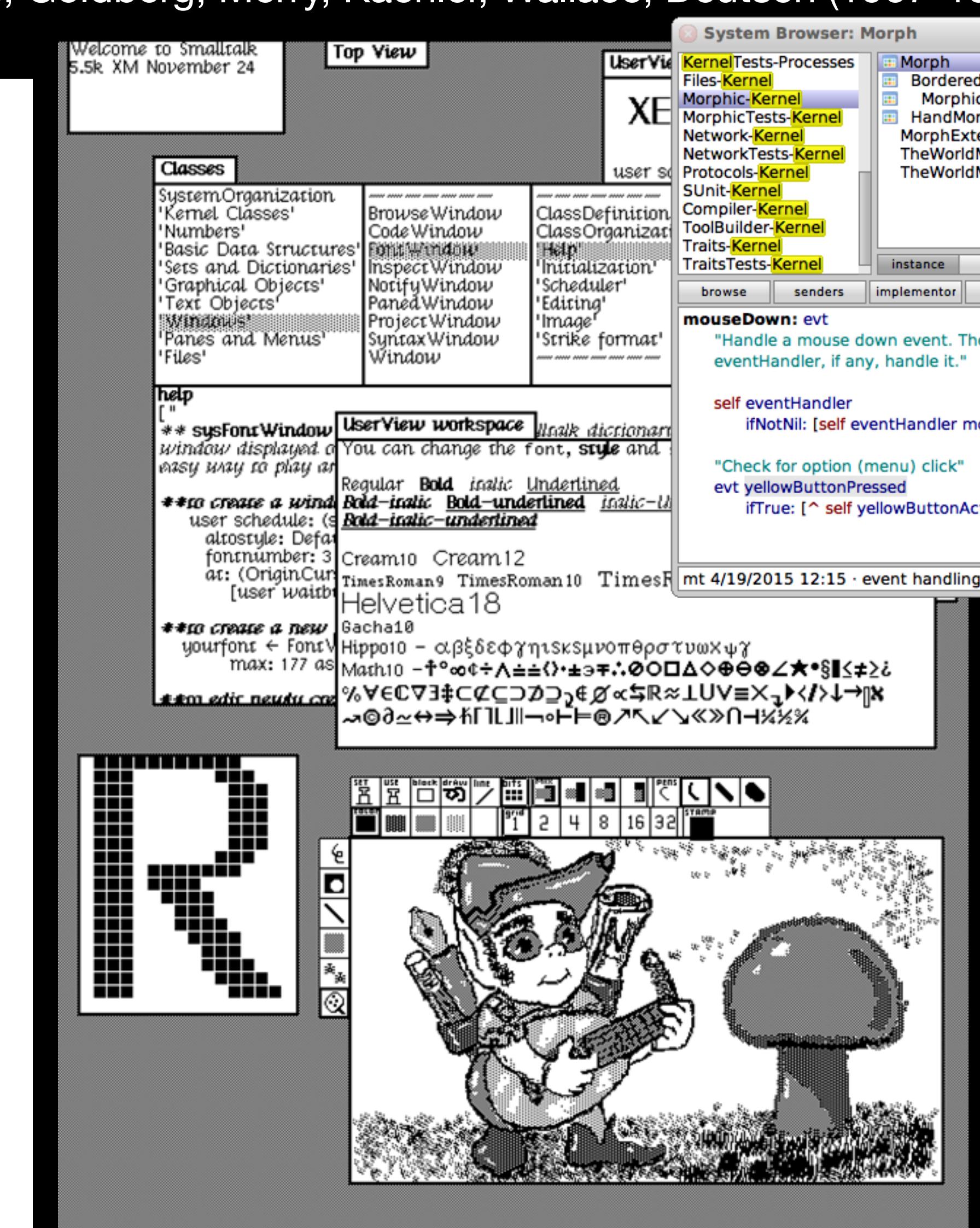
1 to 10!
interval: 1 2 3 4 5 6 7 8 9 10
  

forever!
interval: 1 2 3 4 5 6 7 8 9 10 11 ...
  

1 to 10 do (joe turn 20)!
ok
  

forever do (joe turn 11, jill turn -13)!
ok


```



Pharo
2017

Squeak 2016

Implementors of mouseDown: [65]

- AbstractResizerMorph mouseDown: {as yet unclassified} {Morphic-Windows}
- AlternatePluggableListMorphOfMany mouseDown: {event handling} {Morphic-Pluggable Widgets}
- BookPageThumbnailMorph mouseDown: {event handling} {MorphicExtras-Books}
- BracketSliderMorph mouseDown: {event handling} {Morphic-Widgets}
- ButtonProperties mouseDown: {events} {Etoys-Buttons}
- ChatButtonMorph mouseDown: {event handling} {Nebraska-Audio Chat}

mouseDown: evt

"Handle a mouse down event. The default response is to let my eventHandler, if any, handle it."

self eventHandler

ifNotNil: [self eventHandler mouseDown: evt from: self]

"Check for option (menu) click"

evt yellowButtonPressed

ifTrue: [^ self yellowButtonActivity: evt shiftPressed]

Senders of mouseDown: [37]

- FatBitsPaint toolsForSelection initialization {MorphicExtras-AdditionalWidgets}
- InterimSoundMorph mouseDown: {event handling} {MorphicExtras-SoundInterface}
- MagnifierMorph mouseDown: {event handling} {MorphicExtras-Demo}
- MenuItemMorph mouseDown: {events} {Morphic-Menus}
- Morph click {event handling} {Morphic-Kernel}
- Morph handleMouseDown: {events-processing} {Morphic-Kernel}
- Morph showActions {meta-actions} {Morphic-Kernel}
- NumericReadoutTile mouseStillDown: {event handling} {Etoys-Scripting Tiles}
- PasteUpMorph mouseDown: {event handling} {Morphic-Worlds}
- PluggableListMorph mouseDown: {events} {Morphic-Pluggable Widgets}
- PluggableListMorphOfMany mouseDown: {event handling} {Morphic-Pluggable Widgets}
- PluggableListMorphOfMany mouseMove: {event handling} {Morphic-Pluggable Widgets}

mouseDown: evt

evt yellowButtonPressed

ifTrue: [self chooseMagnification: evt]

ifFalse: [super mouseDown: evt]

bf 9/21/1999 10:45 · event handling · 65 implementors · in no change set ·

World

- System Browser
- Workspace
- Test Runner
- Monticello Browser
- Tools**
- Windows
- Recent Submissions
- Script Manager
- Transcript
- Save
- Save as...
- Save and quit
- Quit

superclass for performance reasons.

index.

tindex]

value: (array at: index).

:= index + 1]

collect:

- collect:from:to:
- collectthenSelect:
- do:
- reverseDo:
- select:
- select:thenCollect:
- sort:
- sort:
- with:collect:
- withIndexCollect:

Interval

OrderedCollection

SharedQueue

*RoelTyper

*Tools-Inspector

*nile-base

accessing

adding

converting

copying

enumerating

removing

sorting

splitJoin

testing

Welcome to Pharo

Recover lost changes...

OrderedCollection

1972

Capabilities

- > "POLA" / Principle of least privilege (Saltzer 1975)
- > "Don't separate designation from authority" (Hardy 1988)

GNOSIS (1972)

KeyKOS (1975)

- > keys (capabilities)
- > single level memory
- > continuous checkpointing + CoW
- > stateless microkernel
- > domains ("actors")
- > segments ("memory")
- > meters ("power supply")
- > resume keys
- > directories map names to keys
- > sensory keys
- > No: shell, fs, address space, callstack, syscalls, priorities, debugging

E
(Miller 1997)

seL4
(Liedtke 2009-)

- > Cap Microkernel
- > Formally verified
- > 10kLoC C
- > ARM, x86, RISC-V

Pony
(Clebsch,
Drossopoulou 2015-)

Fuschia
(Google 2016-)

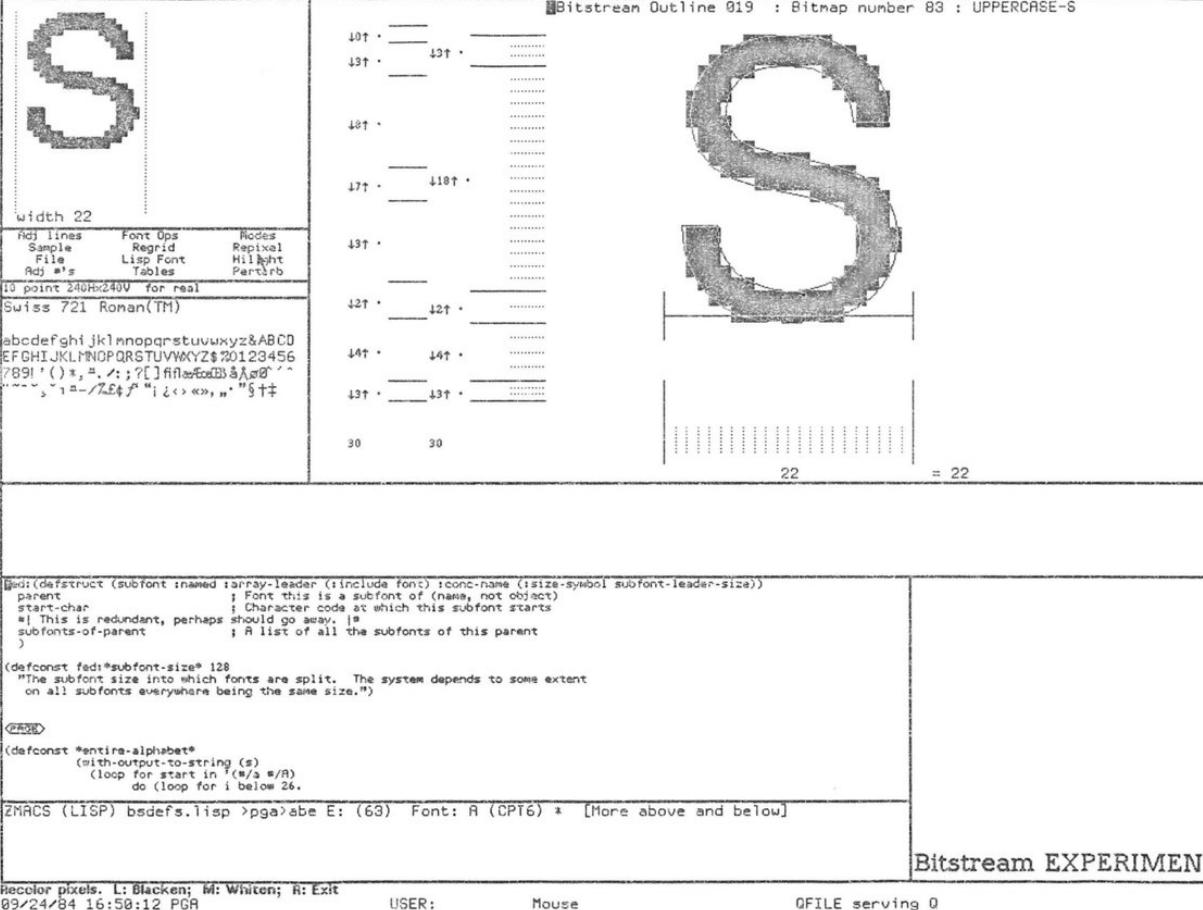
Programming semantics for multiprogrammed computations (Dennis, Van Horn 1966)

Notes on the Confinement Problem (Lampson 1973)

The Confused Deputy (Hardy 1988)

KeyKOS (Hardy 1985, Rajunas 1986, Bomberger 1992, Landau 1993)

GNOSIS (Frantz, Hardy, Lonekait, Landau 1992)



Inspecting presentation CLI::SELECTABLE-ACTIV
in context CLI::SELECT-KEY
 SI::INPUT-EDITOR
 CLI::SELECT-KEY-OR-SELECTABLE-ACTIVITY
 SI::INPUT-EDITOR
 (null context)

from presentation CLI::SELECTABLE-ACTIVITY Flavor Examiner,
to context CLI::SELECT-KEY.

--- **Handlers appearing on mouse buttons** ---

Left	:	(priority 1.5) DW::EXPRESSION-IDENTITY (Mouse-L)
Right	:	(priority 0) DW::MENU (Mouse-R); Menu
sh-Left	:	(priority 0) DW::SELECT-EXPRESSION-AND-ACTIVATI
sh-Right	:	(priority 0) DW::SYSTEM-MENU (sh-Mouse-R); Syst
c-Left	:	(priority 2) DW::CLICK-AND-HOLD-MARK-REGION (c-
c-Right	:	(priority 2) DW::MARKING-AND-YANKING-MENU (c-M
m-sh-Right	:	(priority 0) DW::WINDOW-OPERATION-MENU (m-sh-M
s-Right	:	(priority 0) DW::PRESENTATION-DEBUGGING-MENU (:

--- **Handlers which have been shadowed by other handlers** ---

sh-Left	:	(priority 0) DW::SELECT-AND-ACTIVATE (sh-Mouse-
		Shadowed-by: (priority 0) DW::SELECT-EXPRESSION-
Left	:	(priority 0) IDENTITY (Mouse-L); Flavor Examini
		Shadowed-by: (priority 1.5) DW::EXPRESSION-IDEI

--- **Handlers appearing in the :MARKING-AND-YANKING menu** ---

Display Debugger on Peek			
Abort	Exit	Edit function	Breakpoints
Proceed	Switch windows	Find Frame	Monitor
Return	Help	Backtrace	Exit traps
Reinvoke	Bug Report	Source code	Call traps
Backtrace		Operation on SI:LISP-TOP-LEVEL1:	
		Clear trap-on-exit for this frame	
		Disassemble the function for this frame	
		Edit this frame's function	
		Reinvoke this frame	
		Return from this frame	
		Set the current frame	
		Set the current frame (detailed)	
		Set trap-on-exit for this frame	
		Show arguments with which frame was called	
		Show this function's argument list	
		Marking and yanking menu	
		Presentation debugging menu	
		System menu	
		Window operation menu	

```
Proceed without any special action
Allow process to continue
Return to Lisp Top Level in Dynamic Lisp Listener 1
Restart process Dynamic Lisp Listener 1



---


SCL:PROCESS-WAIT
 0 ENTRY: 2 REQUIRED, 0 OPTIONAL, REST ARG ;Creating PROCESS::WHOSTATE
 2 PUSH NIL ;Creating PROCESS::ARGUMENTS
 4 START-CALL-INDIRECT #'PROCESS:PROCESS-WAIT
 6 PUSH FP|2 ;PROCESS::WHOSTATE
10 PUSH-INDIRECT #'PROCESS::VERIFY-FUNCTION
 7 PUSH FP|3 ;LISP:FUNCTION
12 PUSH LP|0 ;PROCESS::ARGUMENTS
13 FINISH-CALL-APPLY-4-RETURN
```

```
;; -*- Syntax: ANSI-Common-Lisp; Package: CL-USER -*-

(in-package cl-user)

(defun test (x)
  (let ((y (/ x 9)))
    (print y)
    (print (/ 8 y))))
```

Break:
The current frame is **PROCESS:PROCESS-BLOCK-AND-POLL-WAIT-FUNCTION**
s-A, <**RESUME**>: Proceed without any special action
s-B, <**ABORT**>: Allow process to continue
s-C: Return to Lisp Top Level in Dynamic Lisp Listener 1
s-D: Restart process Dynamic Lisp Listener 1
→ Set Current Frame **SCL:PROCESS-WAIT**

Arguments, locals, and specials
Arg 0 (PROCESS::WHOSTATE): "Await Exposure"
Arg 1 (LISP:FUNCTION): #<Compiled function TV:SHEET-EXPOSED-P 20051107372>
Rest Arg: (#<DYNAMIC-LISP-LISTENER Dynamic Lisp Listener 1 20006402557 deexp
Local 3 (PROCESS::ARGUMENTS): (#<DYNAMIC-LISP-LISTENER Dynamic Lisp Listener

□ Arg 1 (SCL:SELF): #<YNAMIC-WINDOW-PANE Dynamic Window Pane 46 (pane INTERACT of Restore Distribution 1) 1005034263 deactivated
s-A, RESUME: Proceed without any special action
s-B, ABORT: Return to Lisp Top Level in Dynamic Lisp Listener 1
s-C: Restart process Dynamic Lisp Listener 1
→ Execute Proceed Return to Lisp Top Level in Dynamic Lisp Listener 1
Return to Lisp Top Level in Dynamic Lisp Listener 1
Back to Lisp Top Level in Dynamic Lisp Listener 1.

Command: Restore Distribution (keywords) :Menu (to select files to restore [default Yes]) Yes
Error: Attempt to activate #<DYNAMIC-WINDOW-PANE Dynamic Window Pane 46 (pane INTERACT of Restore Distribution 1) 1005034263 deactivated>, which has been sent a :KILL message

(FLAVOR:METHOD :ACTIVATE TV:SHEET :BEFORE)
Arg 0 (SYS:SELF-MAPPING-TABLE): #<Map to flavor TV:SHEET 20000343550>
Arg 1 (SCL:SELF): #<DYNAMIC-WINDOW-PANE Dynamic Window Pane 46 (pane INTERACT of Restore Distribution 1
) 1005034263 deactivated>
s-A, RESUME: Proceed without any special action
s-B, ABORT: Return to Lisp Top Level in Dynamic Lisp Listener 1
s-C: Restart process Dynamic Lisp Listener 1
→ Execute Proceed Return to Lisp Top Level in Dynamic Lisp Listener 1
Return to Lisp Top Level in Dynamic Lisp Listener 1
Back to Lisp Top Level in Dynamic Lisp Listener 1.

Command: Select Activity "Restore Distribution" Main X Screen KRAFTBUCH-HOME:0.0 0 (Genera on Dream Factor y)
Activity (program) to interact with next
Type of input expected: an activity name

There are 20 possible activity names. Do you want to see them all? (Y or N) Yes
These are the possible act

These are the possible active screen elements:	
Converse	FEP-
Distribute Systems	File
Document Examiner	Flav.
Editor	Fram.
Command: Select Activity (Lisp
on sheet [default Main]	Inspect
CH-HOME:0.0 0 (Genera on D	Frame-Up
Command:	Any
	Existing Lisp
	Existing Window
	Plain Window
	Trace & Debug
	Trace
	Debug

File system editing operations

Tree Edit Root Tree Edit Any Tree Edit home dir Lisp Window
Refresh Display Help Exit Local LMFS Operations

⇒ (test 0)
0
Trap: There was an attempt to divide by zero in SYS:RATIONAL-QUOTIENT.

TEST

Arg 0 (X): 0

s-A, **RESUME**: Return a value from the RATIONAL-QUOTIENT instruction
s-B: Retry the RATIONAL-QUOTIENT instruction
s-C, **ABORT**: Return to FS Maintenance Lisp Listener in Fsmaint Interaction Pane 1
s-D: FS Maintenance Top Level
s-E: Restart process Fsmaint Frame 1
→ Show Source Code
Source code for TEST:

```
(defun test (x)
  (let ((y (/ x 9)))
    (print y)
    (print (* 8 y))))
```

→

Lisp Interaction Window

Command: Restore Distribution (keywords) :Menu (to select files to restore [default Yes]) Yes
Error: Attempt to activate #<DYNAMIC-WINDOW-PANE Dynamic Window Pane 46 (pane INTERACT of Restore Distribution 1) 1005034263 deactivated>, which has been sent a :KILL message

(FLAVOR:METHOD :ACTIVATE TV:SHEET :BEFORE)

Arg 0 (SYS:SELF-MAPPING-TABLE): #<Map to flavor TV:SHEET 20000343550>
 Arg 1 (SCL:SELF): #<DYNAMIC-WINDOW-PANE Dynamic Window Pane 46 (pane INTERACT of Restore Distribution 1) 1005034263 deactivated>

s-A, **RESUME**: Proceed without any special action
 s-B, **ABORT**: Return to Lisp Top Level in Dynamic Lisp Listener 1
 s-C: Restart process Dynamic Lisp Listener 1
→ Execute Proceed Return to Lisp Top Level in Dynamic Lisp Listener 1
Return to Lisp Top Level in Dynamic Lisp Listener 1
Back to Lisp Top Level in Dynamic Lisp Listener 1.

Command: Select Activity "Restore Distribution" Main X Screen KRAFTBUCH-HOME:0.0 0 (Genera on Dream Factory)
 Activity (program) to interact with next
 Type of input expected: an activity name

There are 20 possible activity names. Do you want to see them all? (Y or N) Yes.
 These are the possible act Split screen element:

Converse	FEP-	Terminal	Lisp
Distribute Systems	File	Peek	Inspect
Document Examiner	Flav	Edit	Frame-Up
Editor	Fram	Distribute Systems	Document Examiner
		Namespace Editor	Any
Command: Select Activity (Existing Lisp	Existing Window
on sheet [default Main		Plain Window	Trace & Debug
CH-HOME:0.0 0 (Genera on D		Trace	Debug
Command:			

Sat Oct 25 22:12



glenda
Apr 13



[win Newcol Kill Putall Dump Exit]

New Cut Paste Snarf Sort Zerox Delcol]

/Users/rsc/g/acme/exec2 Del Snarf Put | Look Slide+]

Advanced Executing

- can run programs like grep or spell.
- but also programs that know about the acme file system.

/Users/rsc/g/bin/Slide+

adict acme
/usr/local/plan9/bin/adict

win

/Users/rsc/g/bin/Slide+ Del Snarf | Look]

#!/usr/local/plan9/bin/rc

name=\$%
current='\${basename \$name}'
currentx='\${9 grep -n ^\$current([])\\$' index | sed 's/:.*//}'

pagex='\${echo \$currentx + 1 | hoc}'

/Users/rsc/g/bin/Slide Del Snarf | Look]

#!/usr/local/plan9/bin/rc

echo name '\${pwd}^/\$1 >/mnt/acme/\$winid/ctl
echo clean >/mnt/acme/\$winid/ctl
echo get >/mnt/acme/\$winid/ctl

addr ctl event tag

body data rdsel xdata

\$ cat tag

/Users/rsc/g/acme/fs Del Snarf Put | Look Slide+\$ cat

body

File System

- acme originated on Plan 9, so it's a file system

- FUSE lets you mount it on Unix systems too

\$ echo hello from the shell >>body

\$]

/Users/rsc/demo/ Del Snarf Get | Look g main

New Cut Paste Snarf Sort Zerox Delcol]

/mnt/acme/2/-libra Del Snarf | Look Send

\$ If /mnt/acme
10/ 28/ cons label
2/ acme/ index new/
\$ cat /mnt/acme/index
2 48 113 0 0 /Users/rsc/g
/acme/fs Del Snarf Put | Look Slide+
10 45 60 0 1 /Users/rsc/d
emo/-libra Del Snarf | Look Send
28 44 47 1 0 /Users/rsc/d
emo/ Del Snarf Get | Look g main
\$ cd /mnt/acme/2

\$ If
addr ctl event tag
body data rdsel xdata

\$ cat tag

/Users/rsc/g/acme/fs Del Snarf Put | Look Slide+\$ cat
body

File System

- acme originated on Plan 9, so it's a file system
- FUSE lets you mount it on Unix systems too

\$ echo hello from the shell >>body

\$]

/Users/rsc/demo/ Del Snarf Get | Look g main

Plan9

Thompson, Pike+ 1992-2015

```

{ structure-diagram }
+-----+
| 00 01 02 03 | 04 05 06 | 07 08 09 | 10 11 12 13 14 15 | 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 |
+-----+
|           sourcePort           |           destinationPort          |
+-----+
|           sequenceNumber        |
+-----+
|           acknowledgementNumber |
+-----+
| offset | reserved | ecn | controlBits |           window          |
+-----+
|           checksum            |           urgentPointer         |
+-----+
tcp -- Transmission Control Protocol packet header [RFC 793]

{
structure :=
error      = ->[self error: !"structure syntax error near: " , [self contents]]
eol        = '\r'`\n'* | '\n'`\r'* 
space      = [ \t]
comment    = [-+] (!eol .)* eol
ws         = (space | comment | eol)*
= space*
letter     = [a-zA-Z]
digit      = [0-9]
identifier = id:(letter (letter | digit)*) _ -> [id asSymbol]
number     = num:$digit+ _
columns   = '|'
( _ num:number
  (num:number)* '|'

ROW        = eol ws
= ( n:number
  ) ? '|'

  _ ( id:identifier '||'
    )+ eol ws
-> (structure-field self id)
-> (set row [row + width])
-> (structure-end id)

name      = id:identifier (!eol .)* eol
diagram   = ws columns row+ name | error
}

'{ svc    = &->(svc? [self peek])
syn    = &->(syn? [self peek]) . ->(out ack-syn -1 (+ sequenceNumber 1) (+ TCP_ACK TCP_SYN) 0)
req    = &->(req? [self peek]) . ->(out ack-psh-fin 0 (+ sequenceNumber datalen (fin-len tcp))
(+ TCP_ACK TCP_PSH TCP_FIN)
(up destinationPort dev ip tcp
(tcp-payload tcp) datalen))
ack    = &->(ack? [self peek]) . ->(out ack
;

(svc (syn | req | ack | .) | . ->(out ack-rst
)
} < [NetworkPseudoInterface tunnel: '/dev/tun0' from: '10.0.0.1' to: '10.0.0.2']

```

COLA - Combined Lambda/Object Architecture

OMeta Piumarta 2007 STEPS Toward The Reinvention of Programming: A Compact And Practical Model of Personal Computing As A Self-Exploratorium

Piumarta 2012

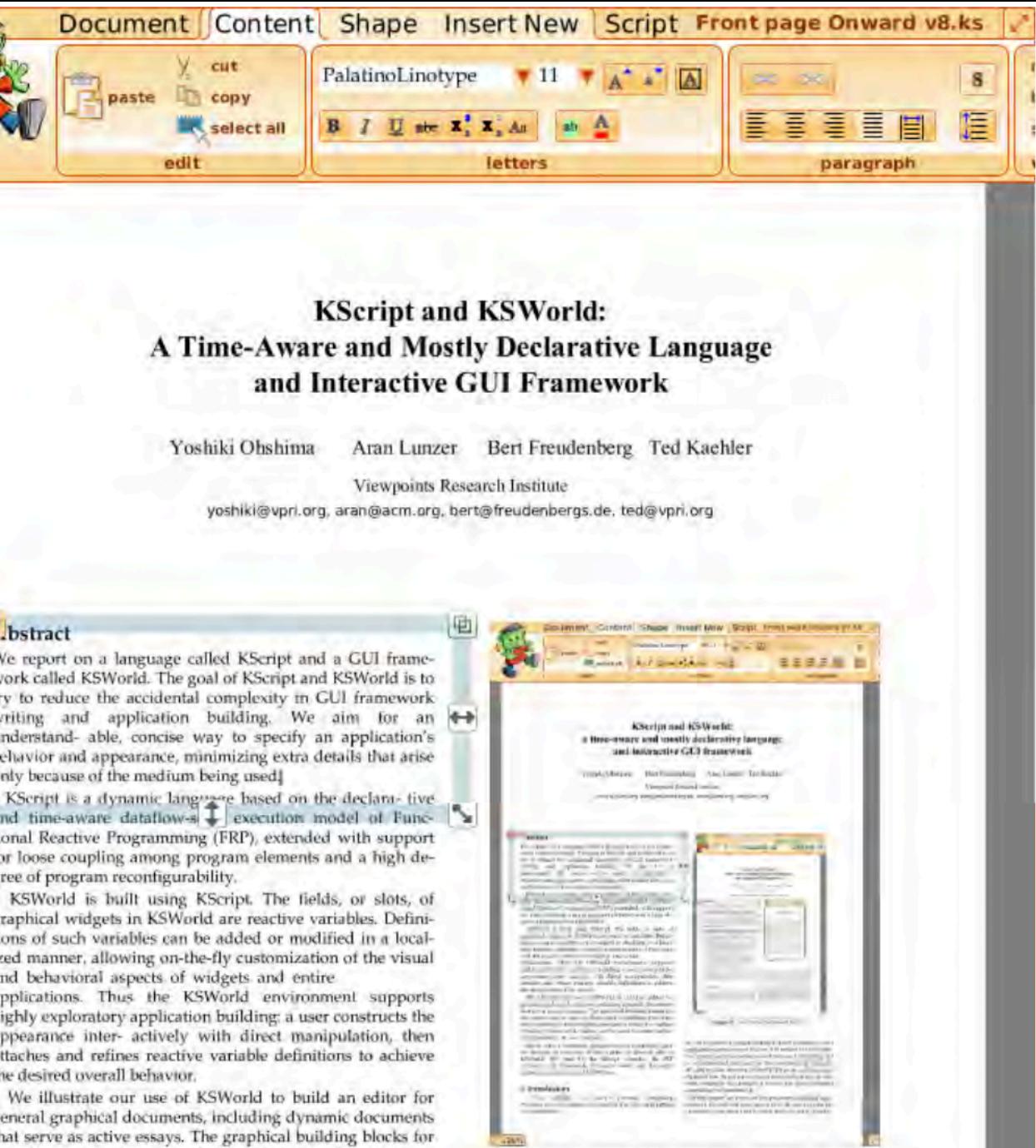
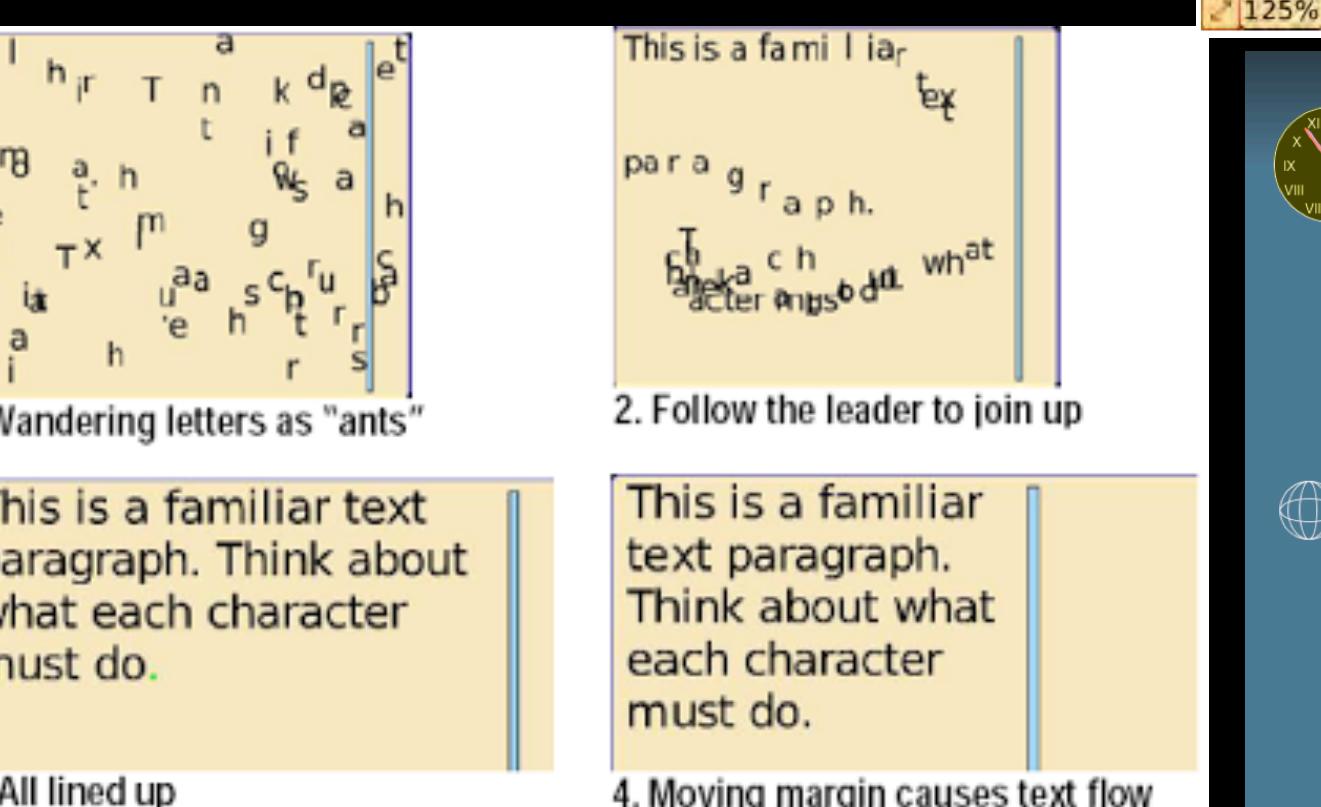
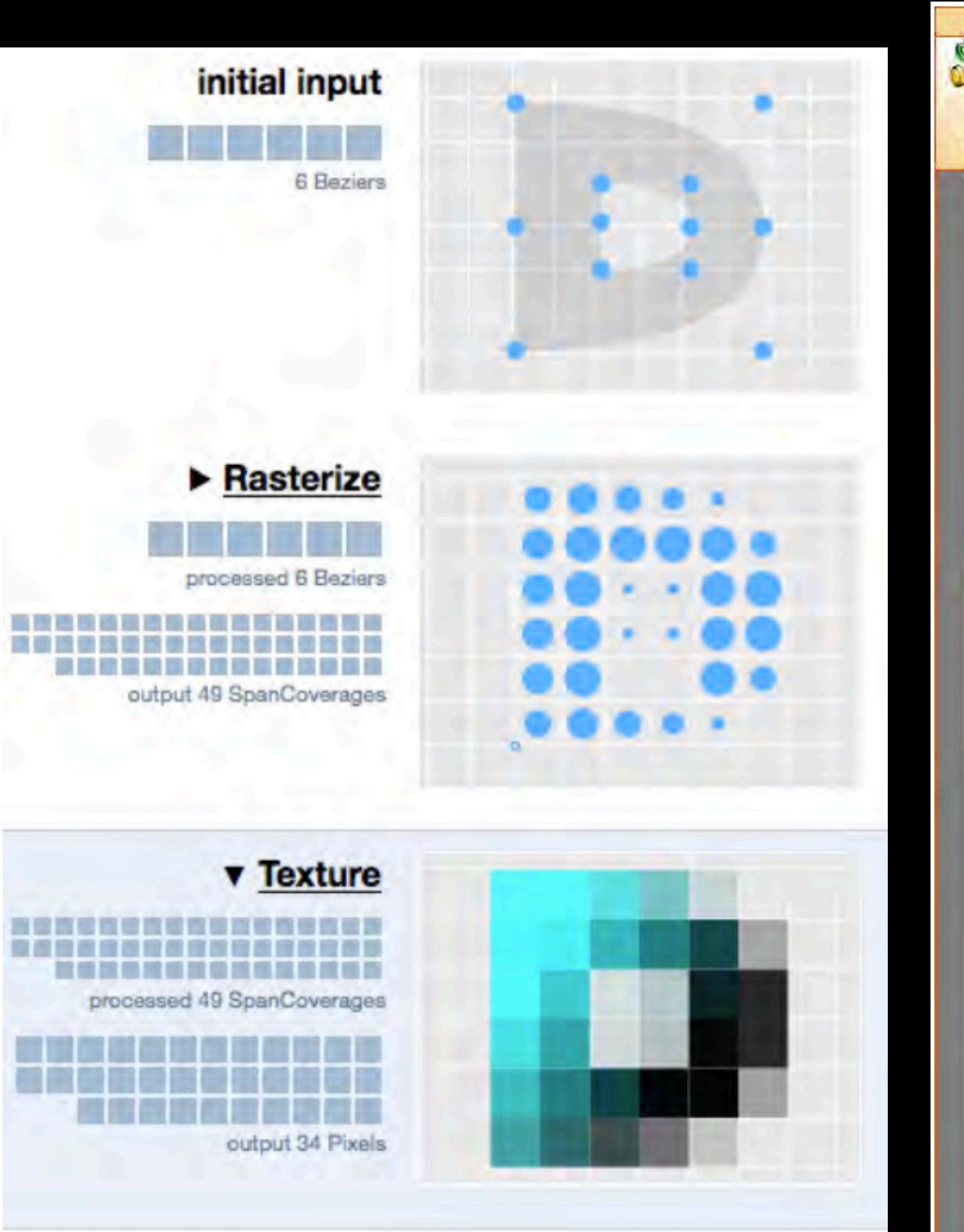
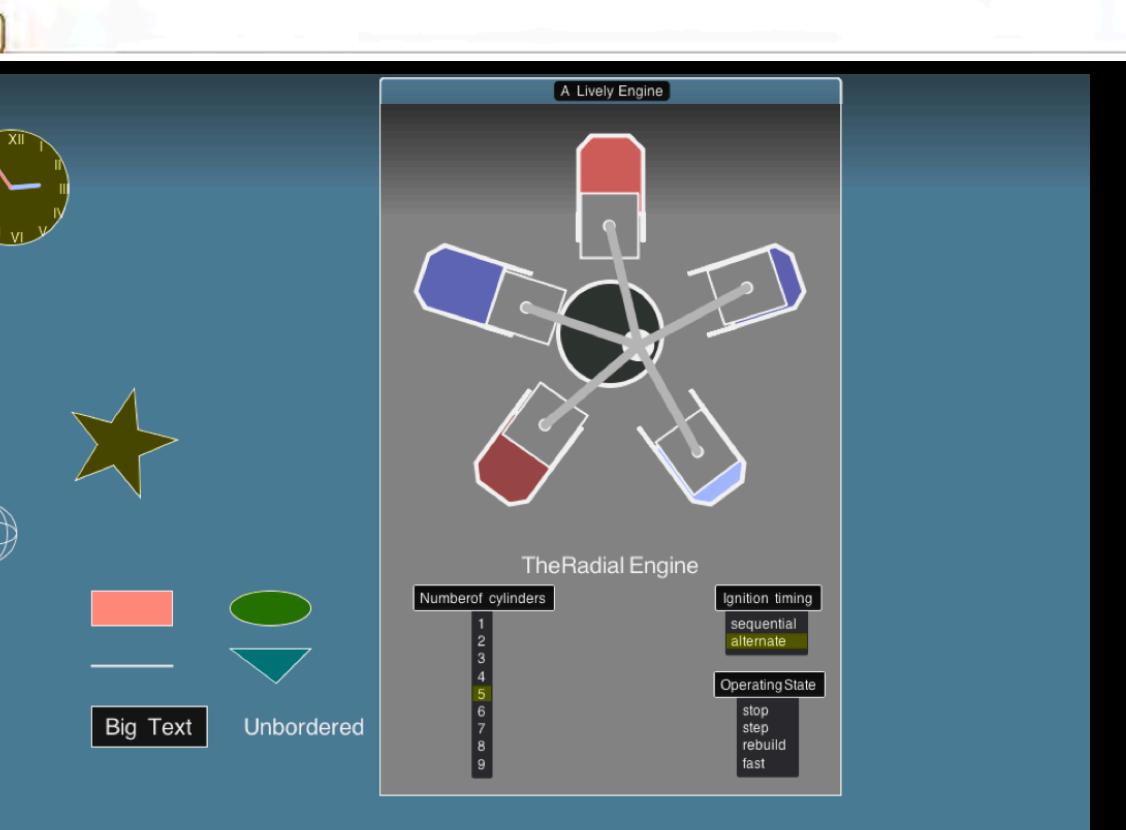


Figure 1. The Frank Document Editor.

an entire system: a typical desktop OS and commonly used application suite amount to over 100 million lines of code. Our group's early experiences with personal computing led us to understand that much of this complexity is "accidental", rather than inherent. In the STEPS project we therefore explored how to reduce such accidental complexity in software, setting as our domain of interest the entire personal computing environment [1].

In this paper we focus on the end-user authoring environment. We feel that end-users should be able to make applications of the same kind as those they are using. Toward



Kay, Piumarta, Ingalls 2008-2012

Contact Application

The App Record

Main App

Pages

About Page
Messages Page
Threads Page
Contacts Page
More Page

Events

Set current page
Display contact
Display messages
Send a message
Archive a thread

Test Data

#app stores some useful state for the application, including the current user, the current page, and the current contact.

```
commit  
  [#app]
```

Set some initial state on #app and mark it as #init. This block sets the start time of the app, so #time is brought into the search. Since we only want this to happen once, we search for #app that is not #init. Then, we mark the #app as #init to prevent the block from firing again due to a time update.

```
search  
  app = [#app]  
  [#time day month year hours minutes seconds ampm]  
  not(app = [#init])  
  user = [#user name: "Corey Montella"]  
  //user = [#user name: "Eric Hoffman"]  
  
  commit  
    app <- [#init user  
      page: [#about contact: user]  
      start-time: [#start-time day month year hours minutes  
      seconds ampm]]
```

Main App

The main application is a shell for all other pages in the app. This component draws the main interface, navigation buttons, and provides a "content" div into which other pages are injected.

```
search  
  app = [#app user]  
  [#time time-string]
```

Eve (witheve.com) Granger 2015-2018

The App Record

Flapping the player

When a player clicks during gameplay, we give the bird some lift by setting its velocity.

```
search @event @session  
  [#click element: [#world]]  
  [#world screen: "game"]  
  player = [#player #self]  
  
  commit  
    player.velocity := 1.17  
  
  0.05% of total | average 0.05ms
```

Scroll the world

Next, we scroll the world in time with frame updates. Eve is currently locked to 60fps updates here, but this will probably be configurable in the future. Importantly, we only want to update the world state once per frame, so to ensure that we note the offset of the frame we last computed in world.frame and ensure we're not recomputing for the same offset.

```
search @session @event  
  [#time frames]  
  world = [#world screen: "game" frame != frames gravity]  
  player = [#player y velocity]  
  not([#click])  
  
  commit  
    world.frame := frames  
    world.distance := world.distance + 1 / 60  
    player <- [y: y - velocity, velocity: velocity + gravity]  
  
  4.30% of total | average 0.11ms | max 0.76ms
```

Drawing

Player

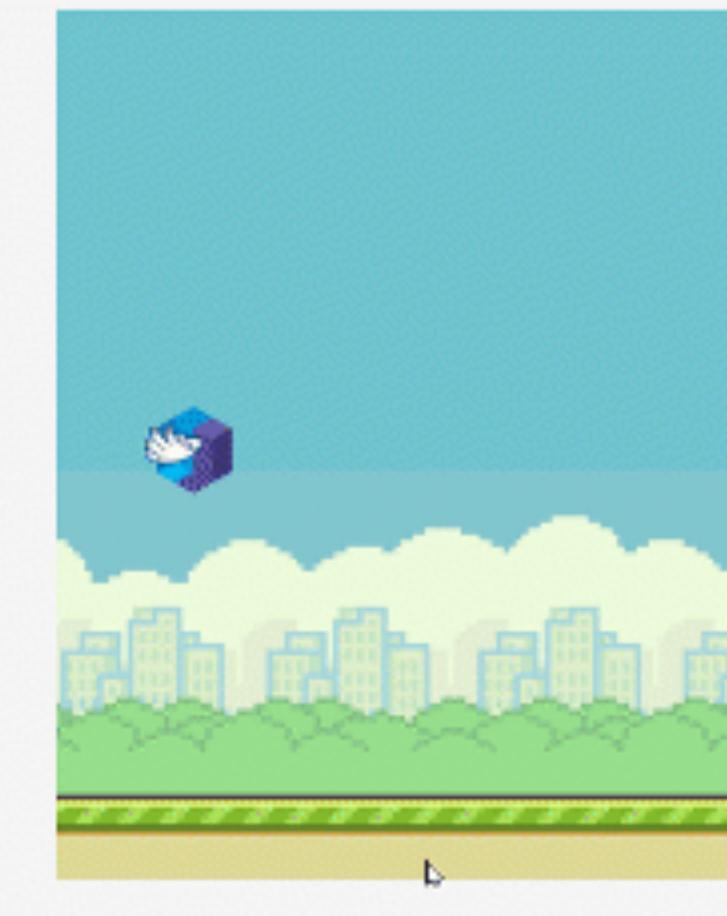
Now we draw the #player at its (x,y) coordinates. Since the player is stationary in x, setting his position here dynamically is just a formality, but it allows us to configure his position on the screen when we're done. We create the sprite first, then set the x and y positions to let re-use the same element regardless of where the player is.

Draw the player

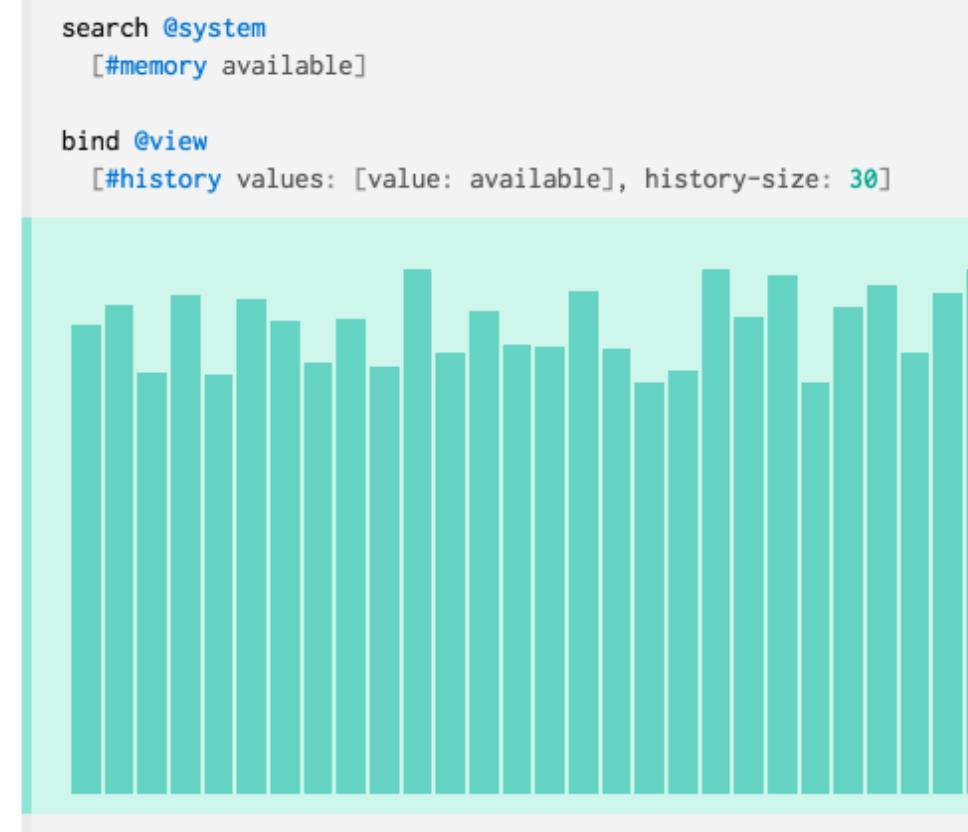
```
search @action @shooter  
  avg = [#game-world]  
  player = [#player x y]  
  
bind @player  
  sprite = [#image player | width: 38 height: 38 href:  
  "http://23.105.104.138:8080/game/world.png?x=8"]  
  position = x: 5  
  speedX = y: 5  
  img.children += sprite
```

Obstacles

Drawing obstacles is much the same process as drawing the player, but we encapsulate the



Graph memory over time



$\frac{\text{quote} \notin \text{dom}(\rho)}{\rho \vdash (\text{quote } d) \Downarrow d}$	(QUOTE)
$\rho \vdash e^* \Downarrow v^*$	
\dots	
$\frac{\text{list} \notin \text{dom}(\rho)}{\rho \vdash (\text{list } e^* \dots) \Downarrow v^* \dots}$	(LIST)
$\frac{(x : v) \in \rho}{\rho \vdash x \Downarrow v}$	(VAR)
$\frac{\lambda \notin \text{dom}(\rho)}{\rho \vdash \lambda x.e \Downarrow < \lambda x.e \text{ in } \rho >}$	(ABS)
$\rho \vdash e_1 \Downarrow < \lambda x.e \text{ in } \rho_1 >$	
$\rho \vdash e_2 \Downarrow v_2$	
$\frac{\rho_1, (x : v_2) \vdash e \Downarrow v}{\rho \vdash (e_1 e_2) \Downarrow v}$	(APP)

Figure 1: Environment-passing Interpreter
(shadowing allowed)

```
(define (eval-expo ρ expr v)
  (fresh ()
    (absento 'in expr)
    (matche expr
      ((quote ,datum)
       (not-in-domo 'quote ρ) (≡ datum v))
      ((list . ,e*)
       (not-in-domo 'list ρ) (eval-listo ρ e* v))
      (,x (symbolo x) (lookupo '(,x : ,v) ρ))
      ((λ (,x) ,e)
       (not-in-domo 'λ ρ) (≡ '((λ (,x) ,e) in ,ρ) v))
      ((,e1 ,e2)
       (fresh (x e ρ1 v2)
         (eval-expo ρ e1 '((λ (,x) ,e) in ,ρ1))
         (eval-expo ρ e2 v2)
         (eval-expo '((,x : ,v2) . ,ρ1) e v)))))))
```

Figure 2: Environment-passing Interpreter
(shadowing allowed, miniKanren)

```
(run99 (q) (evalo q '(I love you)))
(list 'I 'love ((λ (-0) (-0 '-1)) (λ (-2) 'you))),
```

```
(caar (run1 (q) (evalo q q))) ⇒
((λ (-0) (list -0 (list 'quote -0)))
 '(λ (-0) (list -0 (list 'quote -0))))
```

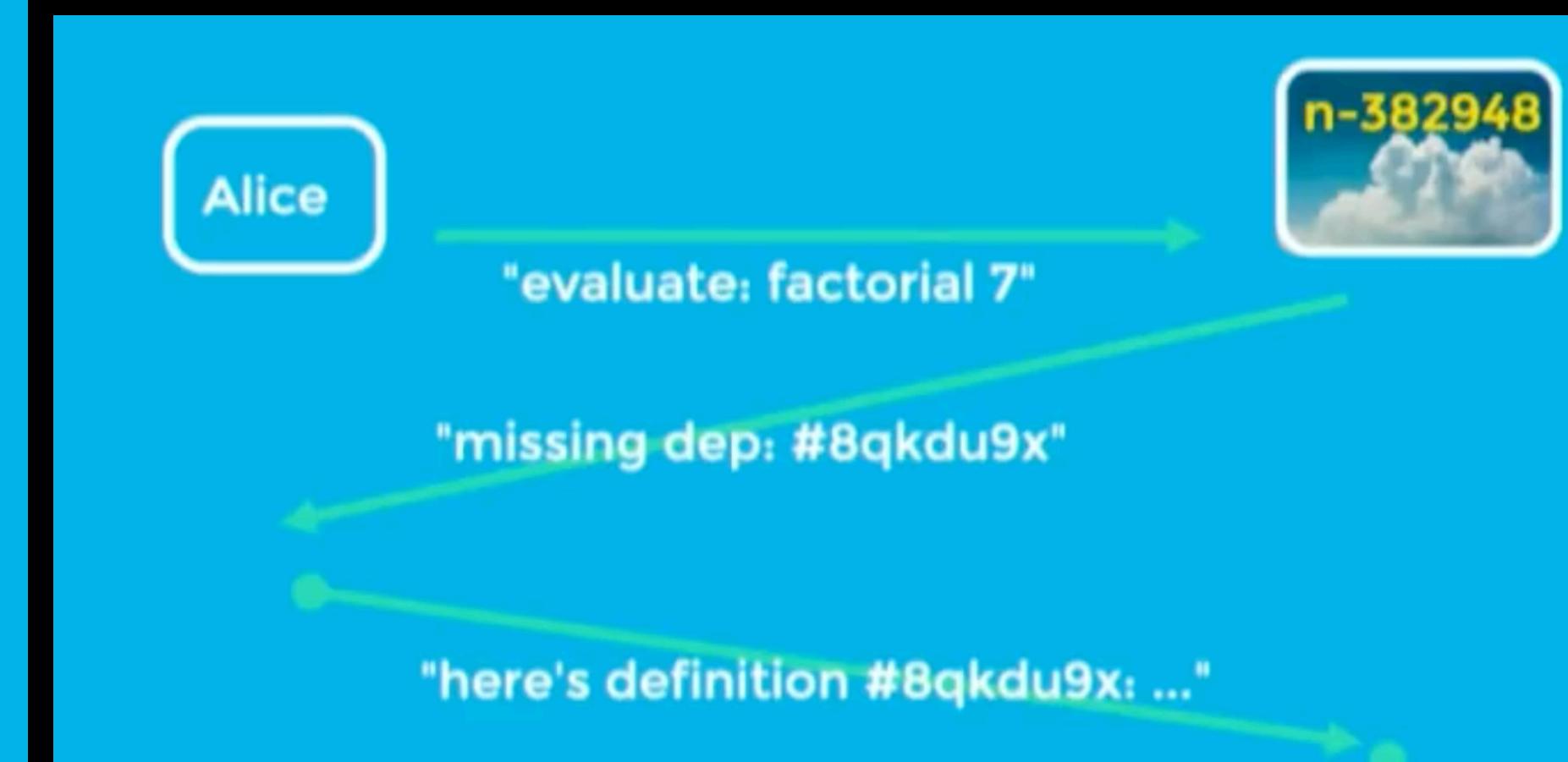
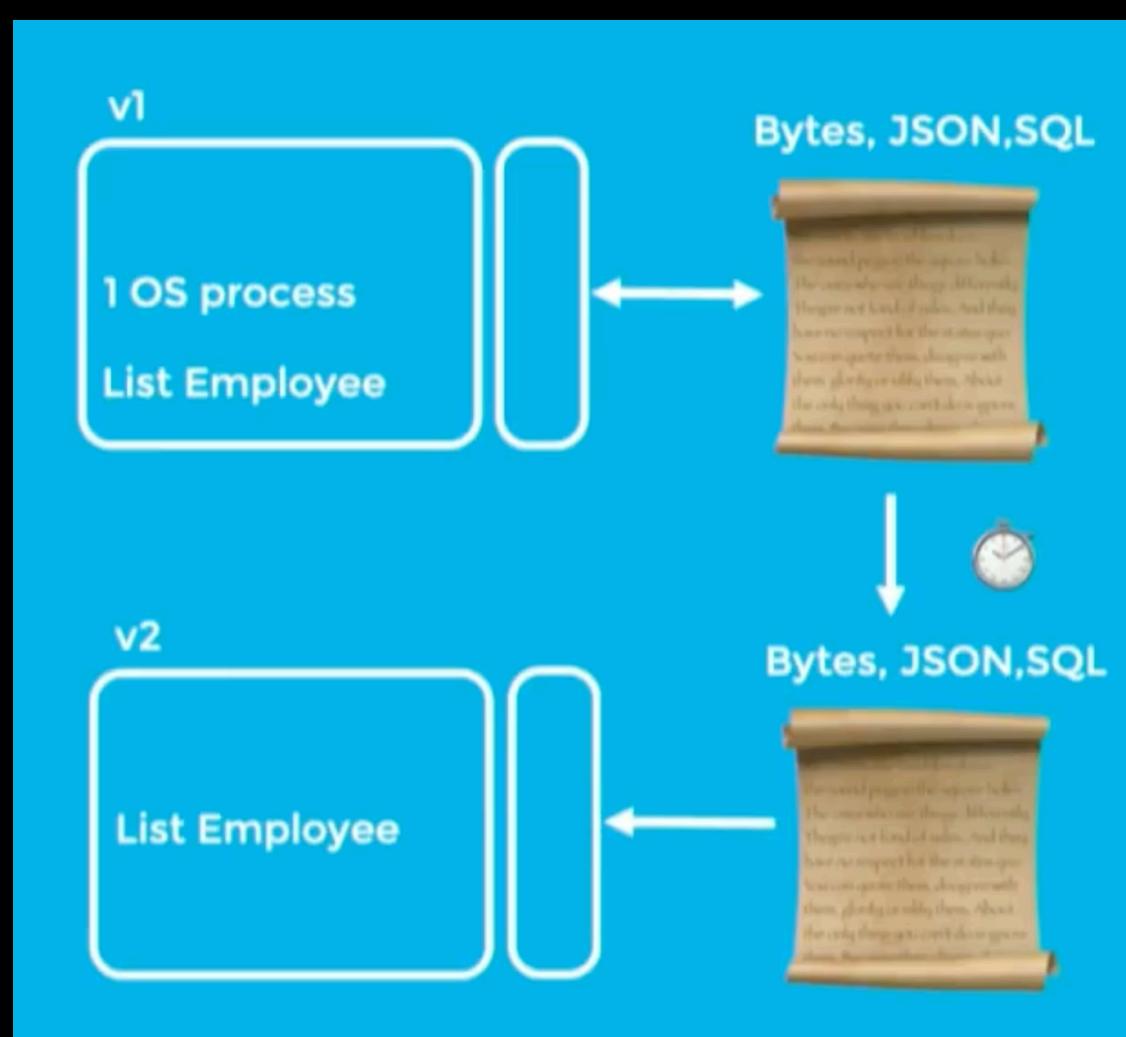
```
(run1 (x)
  (fresh (p q r)
    (≠ p q) (≠ q r) (≠ r p)
    (evalo p q) (evalo q r) (evalo r p)
    (≡ ',(p ,q ,r) x)))
```

miniKanren (Byrd, Friedman 2005)

```
::mouse-hovers-over-text
[:what
  [::window ::width window-width]
  [::window ::height window-height]
  [::mouse ::x mouse-x]
  [::mouse ::y mouse-y]
  [::tab ::current tab-id]
  [::font ::multiplier font-multiplier]
  [tab-id ::left left]
  [tab-id ::right right]
  [tab-id ::top top]
  [tab-id ::bottom bottom]
:when
(<= left mouse-x (- window-width right))
(<= (top window-height font-multiplier)
  mouse-y
  (bottom window-height font-multiplier))
:then
(→ o/*session*
  (o/insert ::mouse {::target :text
    ::cursor :ibeam})
  o/reset!)]
```

Paravim (Oakes 2019)

- > Functions+values+partially applied fns stored in db, indexed by hash of ast
- > Append-only (immutable) (+ x 2) == (+ x 1 1)
- > No builds
- > Trivial renames
- > No dependency conflicts
- > Trivial caching: ast, type info, unit tests
- > Typed durable storage
- > "Programs that deploy themselves and describe whole elastic distributed systems"
- > Scratch file + codebase manager tool: watch expressions (replaces REPL)
unit tests
add, rename, view
- > Algebraic effects ([Kammar, Lindley, Oury 2013](#))



```

factorial n = product (range 1 (n + 1))

factorial =
  n → #8h7jbs5 (#h9bxw9q 1 (#e2i3biw n 1))

factorial =
  $arg1 → #8h7jbs5 (#h9bxw9q 1 (#e2i3biw $arg1 1))

```

```

1 dsort : (a → a → Boolean) → [a] →{Remote} [a]
2 dsort lte as =
3   if size as < 2 then as
4   else case halve as of (left, right) →
5     resL = at spawn '(dsort lte left)
6     resR = at spawn '(dsort lte right)
7     merge lte (force resL) (force resR)

```

```

1 ability Remote where
2   at : Node → '{Remote} a → Future a
3   force : Future a → a
4   spawn : Node

```

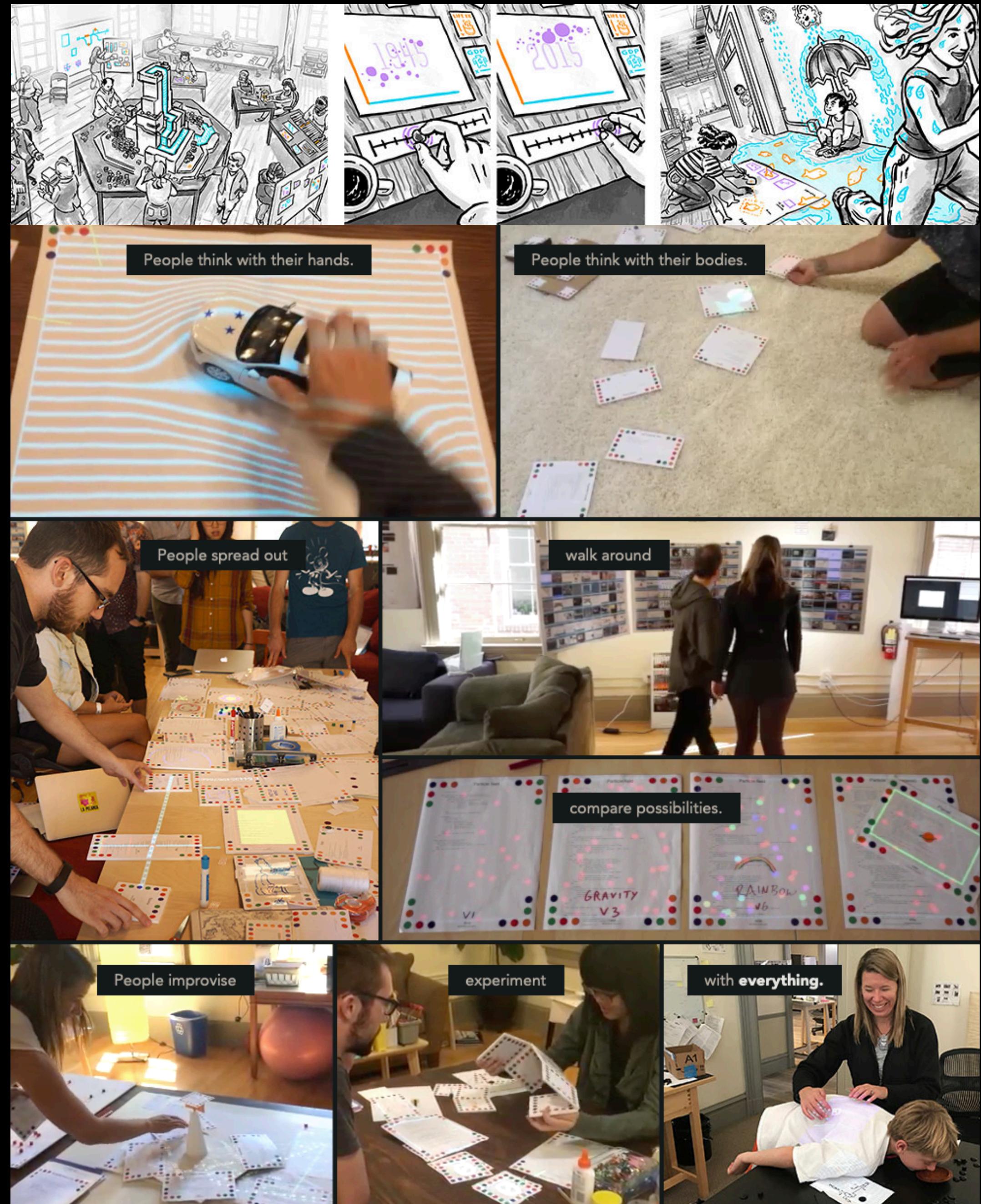
```

1 Remote.runLocal '(dsort hugeDataset)
2
3 Cloud.usEast.run '(dsort hugeDataset)

```


- coding → direct manipulation of data
- procedures → goals and constraints
- text dump → spatial representation
- sequential → concurrent

Victor 2012



Dynamicland
Victor 2018

TUNES
Rideau 1992-1995

Nganhm the Houyhnhnm Computing
Rideau 2015-2020

"Multix"
Gottschalk 2017-2020

Intel iAPX 432 / i960 (1981-1986)

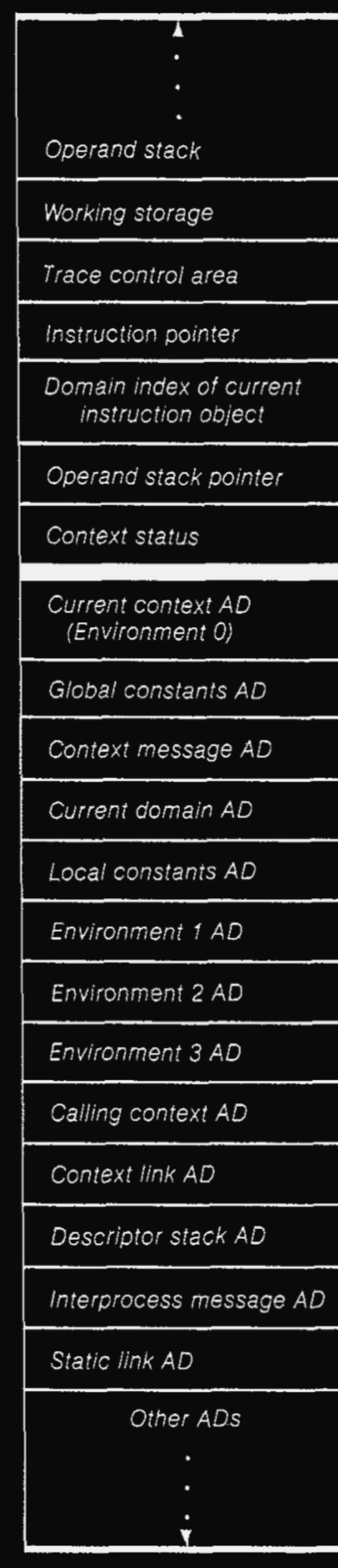
- > Clean slate "Micromainframe"
- > Microcoded ISA
- > "Object-oriented" (Gehringer, Colwell 1986)
- > Instructions to access abstract data types
- > Tagged pointers
- > POLA Segments
- > Microcode garbage collection
- > Object Capabilities (Levy 1984)
- > OS in Ada

But: (Smotherman 2010)

- > x0.25 performance of comparable chips
- > Bit-aligned variable length instructions
- > 2-chip implementation
- > "Fault tolerance" 40% idle
- > Expensive high level Instructions



Data Part



Access Part

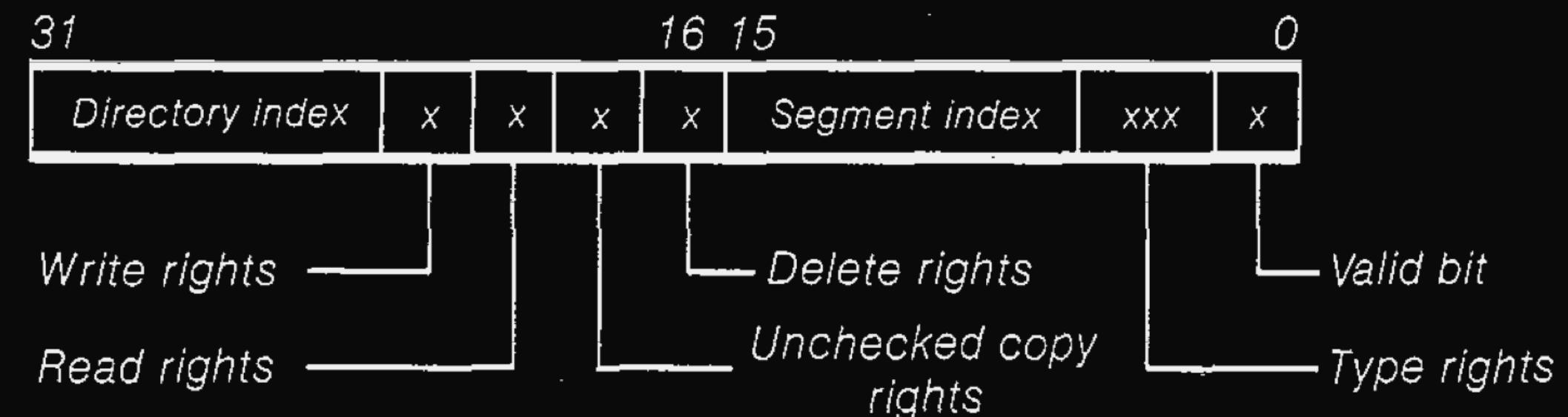


Figure 9-4: Intel 432 Access Descriptor

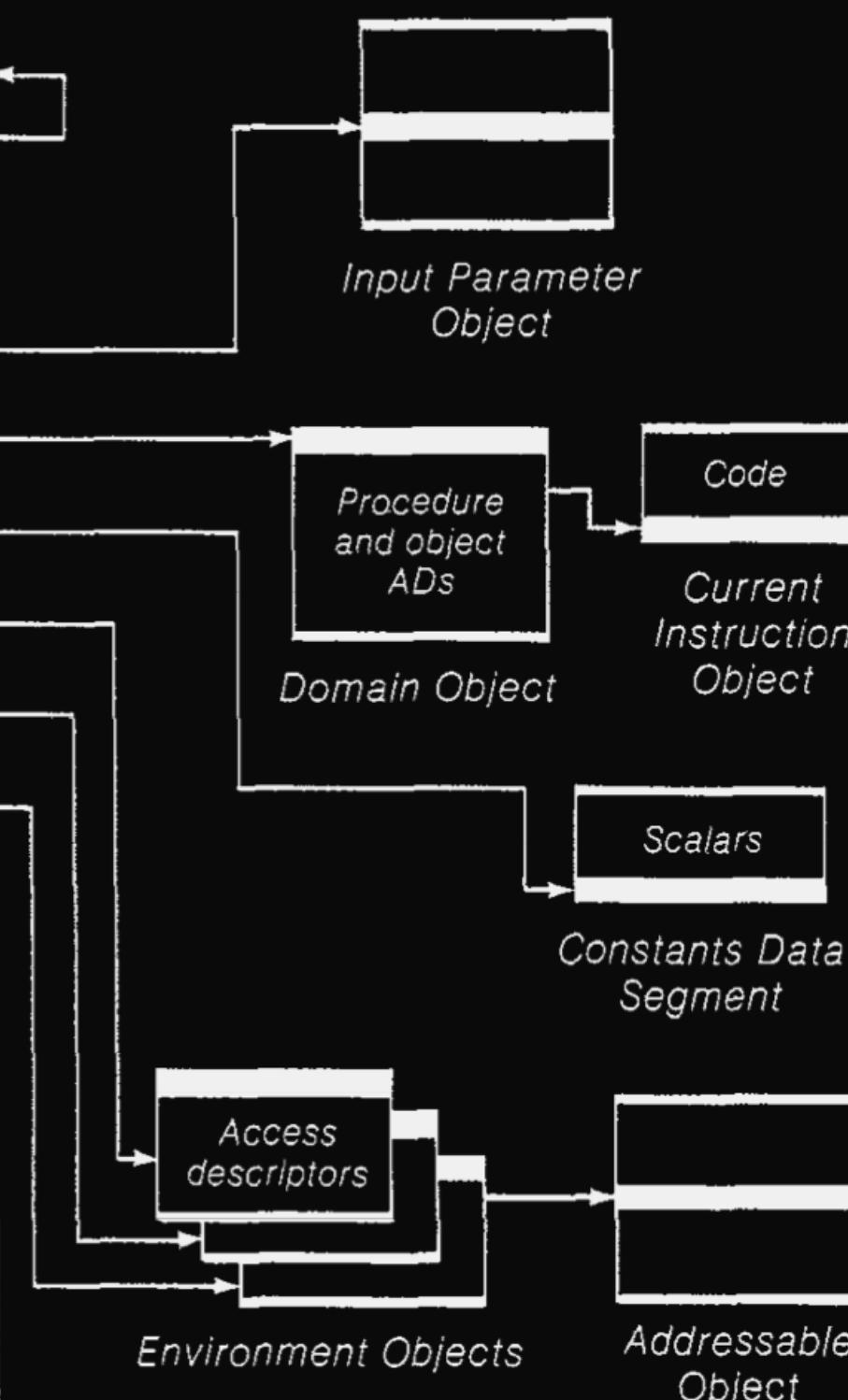


Figure 9-7: Intel 432 Context Object Representation

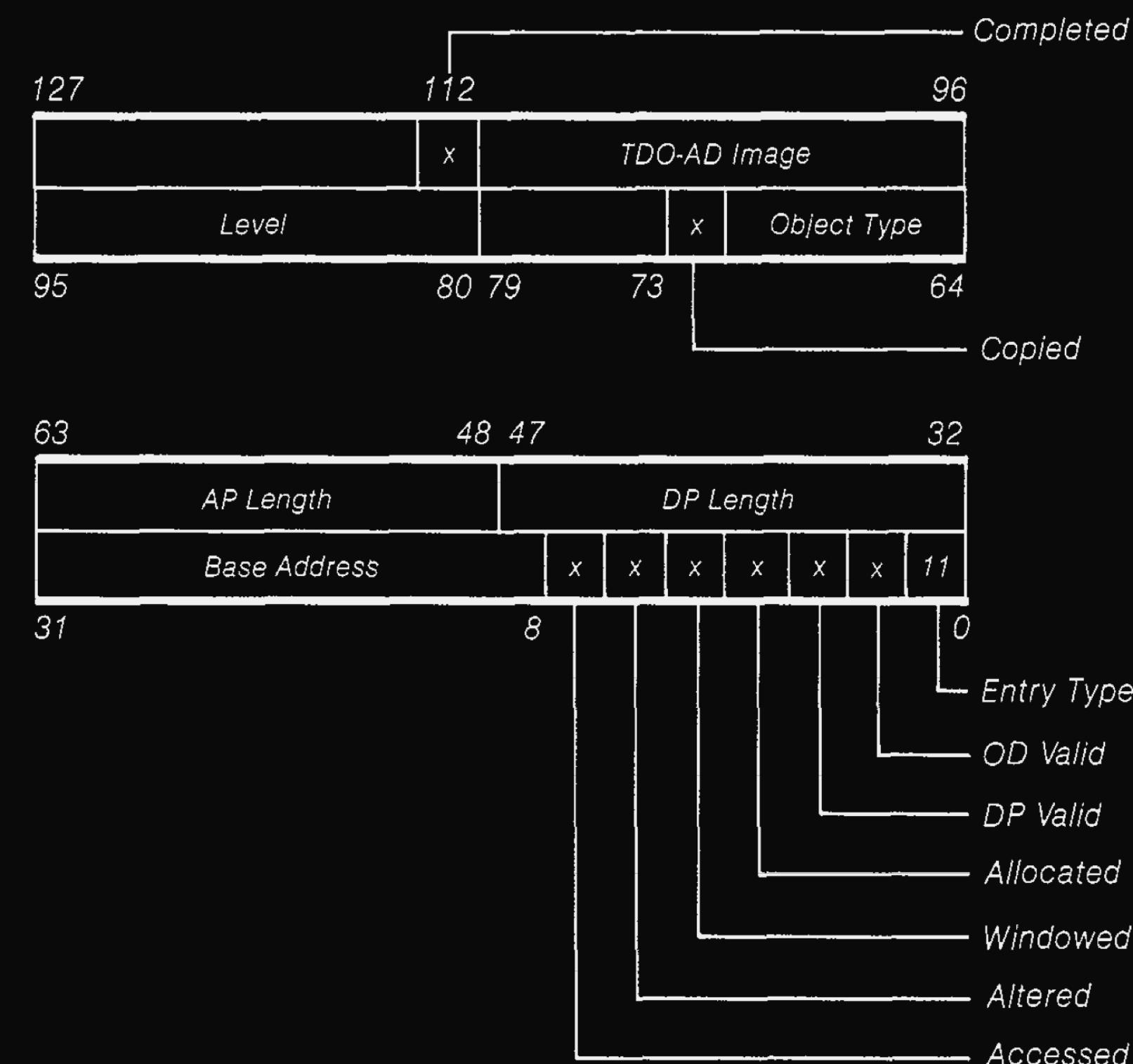


Figure 9-3: Intel 432 Storage Segments Descriptor