

The Function Final Frontier



	Sustan Braunas				00	
	Collections Con				127 (
	Collections-Sequer	Interval	accossing	Collect		
	Collections-Arrayo	Linkodlist	conving	do:	all a s	
	Collections-Stream	MannedCollection	adding	do:andBetweenDo:		
	Collections-Suppor	OrderedCollection	removing	promoteFirstSuchT	1234	
	Graphics-Primitives	SortedCollection	enumerating	reverse	12 14 15 17 15 14 21	
	Graphics-Display C		private	reverseDo:	22 23 25 28 27 29 30	
	Graphics-Media			select: Form Editor		
	Graphics-Paths	instance class	1			
	collect: aBlock	instance class			tina	
	"Evaluate a Blo	ck with each of my	elements as the a	comment. c	10.1.	
	resulting values into a collection that is like me Answer with					
	resurring variaes	rido annoration il	and to the me. An	and advant.		
	collection. Over	riae superciass in a	order to use add:,	not at:put:.		
					T	
newCollection						
	newCollection + self species new.					
	self do: [:each newCollection add: (aBlock value: each)].					
	<pre>tnewCollection</pre>					
User	Interrupt					
Paragraph>>characterBlockAtPoint:						
Paragraph>>mouseSelect:to:						
CodeController(ParagraphEditor)>>processRedButton						
CodeController(ParagraphEditor)>>processMouseButtons						
CodeController(ParagraphEditor)>>controlActivity						
5000	controller (controlle	, presidenti oleoop			-max	
controlActivity						
self scrollBarContainsCursor						
ifTrue:						
[self scroll]						
	ifFalso	File List			EX	
	Freif eren	Kauka [](Bobson)	SF>*		E A	
	[self process	Keybo Ljunoboli	A		E	
self processMouseE [Filene](Robson)SF)ScreenForm.st						
		[Filepe]/Bol	son)SE)ScreenFol	mChannes st	=//	
blue	Butto 31@537 corne	r: [Filene](Bol	son)SF)WordGran	bics.form		
scrol	Bar 63@770	[].			THE	
mark	er	Postanzia	from I to relate			
save	dAre	nectangle	fromoser origin	F		
para	grap					
start	Bloc	ScreenForm	setFullPageWidth			
		ScreenForm				
		printRe	ctangle:			
		(30	@5 extent: 674@	790)		
		onFileN	amed: 'ExampleSc	reen.press'		
(Form readFrom: 'FilledSkate.form') edit						
		(Form read	From: 'FilledSkate.1	form") edit		

Model-View-Controller

- first formulated by Trygve Reenskaug
 Adele Goldberg and others at Xerox
 PARC in 1979
- long shadow, the basic concepts still prevalent today. Good ideas eat their children.

- At a very abstract level MVC is a sound separation of concerns
- Implementations leave much to be desired
 - Stateful objects everywhere





Mutable DOM

Functional Programming?

- Functional Reactive Programming (FRP), still active area of research
 - Rx, doesn't address rendering
- Communicating Sequential Processes (CSP), a coordination language, doesn't address rendering

Om

"The importance of the program cannot be overlooked _ it so fundamentally alters the mechanics of mathematics."

Allow York (Select).

'Mathematics has a real chance of replacing pencil and paper as the standard desktop appricament for scientists and angineers." William Press Autoury of Acceptyons, Reveal University 'Mathematica will revolutionize the teaching and learning of math by focusing on the prose of methomatics, without gotting lost in the grammar." Strain Jahr

President Mail Int

Mathematica.

A System for Doing Mathematics by Computer

Some facts:

Function: Numerical, symbolic, graphical computation, interactive programming. Integrated technical computing environment.

Numerical Computation: Arbitrary precision arithmetic, complex numbers, special functions (hypergeometric, elliptic, etc.), combinatorial and integer functions. Matrix operations, root finding, function fitting, Fourier transforms, mamerical integration, minimization, linear programming.

In/1/1= 3*70

2503155504993241601315571986085849

11/2/2= Rypergeometric2F1[7, 5, 4.1, 3-1]

-0.00403761 - 0.00295663 I

Numerical Computation

Symbolic Computation: Equation solving, symbolic integration, differentiation, power series, limits. Algebraic operations, polynomial expansion, factorization, simplification. Operations on matrices, tennors, lists.

Graphics: 2D, 3D plots of functions, data, geometrical objects. Contour, density plots. 3D rendering with intersecting surfaces, lighting models, symbolic descriptions. Color Post-SCOPT output, publication quality graphics, animation (most versions).

Symbolic Computation

Programming: High-level, interactive, symbolic system. Full procedural language, functional programming constructs. General transformation rule paradigm based on pattern matching.

External Interface: Input from external files, programs. Output in TeX, C, FORTRAN, Post-SCRIPT. Interactive external process control (most versions).

Graphics and Visualization

Macintosh User Interface: Notebook interactive documents mixing text, graphics, seimations, Mathematics input, output. Macintosh front end can be used with kernels on other computers. Macintosh graphics standards used.

Doing Mathematics by Computer" by Stephen Wolfram (Addison-Wesley, 1988) available at bookstores. Additional documentation supplied with specific versions. Mathematica Journal to be published in 1990.

Versions New Available: Apple Macintosh \$495 (Plue, SE, etc.); \$795 (II, ID, IICK, SE/30, etc.) + 386-based MS-DOS systems: \$695 (no coprocessor); \$995 (287/387); \$1295 (Weitek) + Apollo DN 2500-4500, 10.000; from \$2400 + Data General AVIION: \$2,800 • DEC VAX VMS, ULTRIX, RISC-based systems: from \$2400 Hewlett-Packard 9000/300, 800: from \$2400 • IBM AIX/RT: \$2400 • MIPS: from \$2800 • NeXT: bundled as standard system software + Silicon Graphics IRIS: from \$2800 + Sony NEWS: from \$2400 • Sun 3, 4, 386c from \$2250 • Supercomputer and other versions also available. • Educational, volume, reseller, and other dis-counts available • Now shipping Version 1.2.

log[1] = 0 log[E] = 1

log[s_"n_] := n log[s]

log/: inverseFunction[log] = exp

Sum[-(-1) *k (x-1) *k/k, (k, 1, a)] +

Nigh-Lovel Programming

Implementation: 770 pre-defined Mathematica functions (C source 190,000 lines). Design, development led by Stephen Wolfram. Version 1.0 released June 1968.

education, mathematical modeling, publication graphics, data analysis, visualization, systems analysis, algorithm development.

Awards: Best 10 New Products, Basiness Week 1988 · Editor's Choice Award, Macliser 1989 · Award of Distinction, JIYTE 1988.

Wolfram Research, Inc.

P.O. Box 6058, Champaign, IL \$1825-6059, USA. Information: 217-395-6050. Ordens: 800-441-MATH. Or visit your local software dealer.

B 100 Worker Reserve in: Advisoration is experient testimeted in terms of Worker Research. In: All other sedematics are taken and a section testimation of the experience research and the experience is an end of the experience of a size end of the testimation of a size end of the testimation.

Documentation: "Mathematica: A System for

log[x_ y_] := log[x] + log[y]

log'[s_] = 1/x (* derivative *)

109/: Series[log[x_], (x_. 1, n_)] :=

0[x, 1]*(n+1)

Typical Applications: Research, engineering,

$f(D_0) = V_0$

$f(D_1) = V_1$

$diff(V_0, V_1) = CHANGES$

Why diffs?

- Views just re-render when data changes
- No explicit observation (and thus no resource issues)
- Much less logic to write around data observation & view updates

demo

Modularity?

OOP

- Objects naturally modular
 - but not modular with respect to state!
- Preserve component modularity
 - But also achieve modularity with respect to to state.

Multi-Stack Environment part 2

App State

- An immutable tree of associative data
- "Global" state
 - It's not as scary as it sounds (like a database!)
- Need local/global to be a point view
 - How?

Cursors

- A triple, data to render (consistent), path into the global app state, reference to global app state
- Track the path via normal collection access patterns (more natural than zippers)

(def app-state
 {:foo {:bar [{:woz ...} ...]}})

(om/root some-view app-state
 {:target ...})

```
:value {:foo {:bar ...}}
  :path []
  :state #<Atom: {:foo ...}>
(defn some-view [data owner]
  (reify
    IRender
    (render [_]
      (let [x (:foo data)]
        ...)))))
```



```
(defn some-view [data owner]
  (reify
    IRender
    (render [_]
      (let [x (:foo data)
            ....]
        (om/build another-view
        (get-in x [:bar 0])))))
```


Modularity regained!

Not enough ...

:instrument

- Aspect Oriented Programming
 - Tasteful application of global concerns
- Can intercept the construction of any component and modify behavior
- Generic editors! (Manipulate)

demo

:tx-listen

- Stream of full app states is not nice for serializing to disk or sending updates to remote servers
- :tx-listen gives path and old and new value at that path
 - like git patches!

Simple and Easy

- Undo/redo, time travel
- Meta components
- Synchronization online/offline
- Om components play well with others

UI Spectrum

session @kovasb

demo

Questions?