A SMALL MATTER OF PROGRAMMING

OF PROGRAMMING

PERSPECTIVES ON END USER COMPUTING
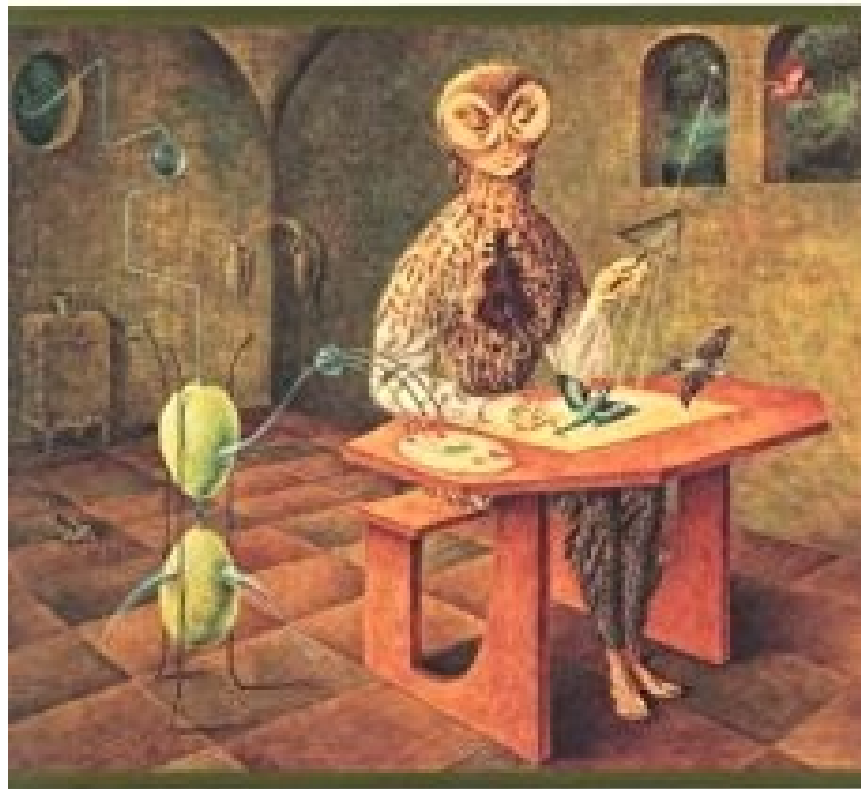
BONNIE A. NARDI

End user	formulas	macro recorder	node editor	Unix shell	level editor
Domain dev	macros	textual macros	scripting	shell script	game scripting
Core app	spreadsheet	word processor	3D app	C utilities	video game

sample-1   sample-2

sensor-patch

<--Render box 1

r sensValueB
0

send messages
FULLSCREEN 1
then
CREATE

create
destroy   CREATE
fullscreen 1
fullscreen 0
gemwin

print b

select 50   trigger bang bang
stop   metro 25
                   float

select -50   trigger bang bang
stop   metro 25
                   float

threshold 1500 0 1501 0   threshold 1500 0 1501 0   threshold 1500 0 1501 0   threshold 1501 0 1
<= 1501   <= 150

Open Video 1-->
Render box 2-->   Render box 3-->   Open Video 2-->
gemhead   openpanel   gemhead   openpanel
auto $1   open $1   auto $1   open $1
pix_film   pix_film

pix_mix 0

Open Sound 1-->   Tabplay bangs must both
                          be manually pressed on
                          first setup
pix_texture
                 openpanel   tabplay- sample-1
rectangle   read -resize $1 sample-1   <-- Make sure this box IS toggled on first set
                 soundfiler   dac

                                  pd dsp 1   <--Press this to compute audio

Open Sound 2-->
                 openpanel   tabplay- sample-2
Set to 4 and 4   read -resize $1 sample-2   <-- Make sure this box is NOT toggled on first
Click and drag   soundfiler   dac
in the boxes

File Render Window Help   Back to Previous   Cycles Render   v2.74 | Verts:37,370 | Faces:37,754 | Tris:74,716 | Objects:1/45 | Lamps:0/1 | Mem:321.86M | Camera

Fichier Édition Affichage Insertion Format Outils Données Fenêtre Aide

0.00cm   Gris   Couleur

K28   Σ =

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Type	2000	2001	Mean 00	Mean 01								
2	Banana	45	79	50	57								
3	Apple	48	75	40	47								
4	Orange	48	48	55	62								
5	Grape	34	48	60	67								
6	Durian	78	45	20	27								
7	Coconut	12	45	70	77								
8	Framboise	48	54	50	57								
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													

Fruits (in percentage)

90
80
70
60
50
40
30
20
10
0

Banana Apple Orange Grape Durian Coconut Framboise

2000
2001
Mean 00
Mean 01

dataflow

dataflow

(functional programming)

PUC-Rio

DEPARTAMENTO DE INFORMÁTICA

PhD Thesis

Dataflow Semantics for End-User Programmable Applications

Author:
Hisham Hashem Muhammad

Advisor:
Roberto Ierusalimschy

January 19th, 2016

the interaction of well-intentioned features
can have very bad results

Pure Data has
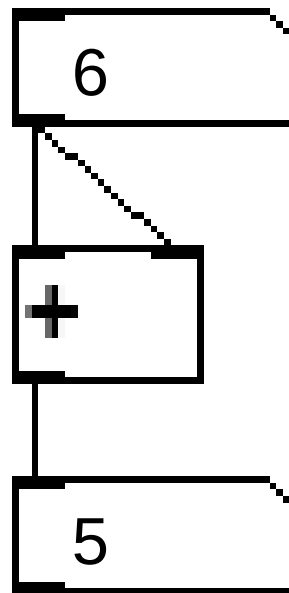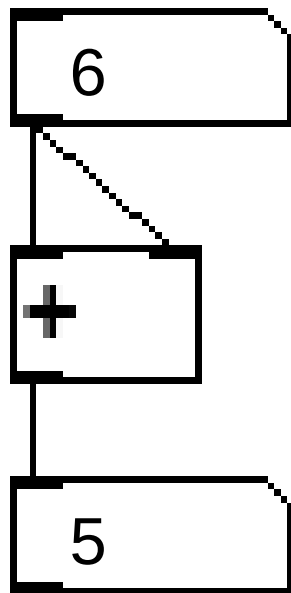
- multiple outputs per connection

Pure Data has

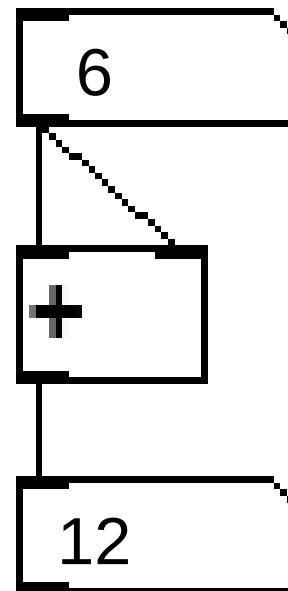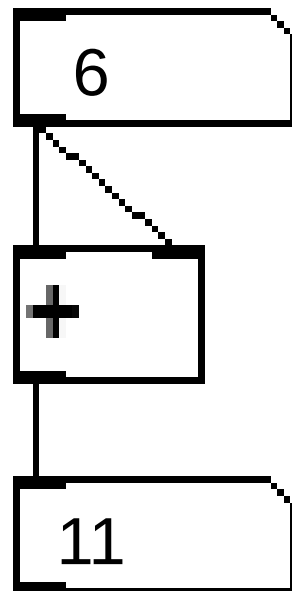- multiple outputs per connection
- deterministic execution

Pure Data has

- multiple outputs per connection

- deterministic execution

- graph loops for building delays and echoes

Pure Data has

- multiple outputs per connection

- deterministic execution

- graph loops for building delays and echoes

- triggering and non-triggering inputs

```
┌──────────────┐          ┌──────────────┐
│  6           │          │  6           │
└──────────────┘          └──────────────┘
    │ ╲                        │ ╲
    │  ╲                       │  ╲
  ┌─┴────────┐              ┌─┴────────┐
  │ +        │              │ +        │
  └──────────┘              └──────────┘
    │                          │
  ┌─┴────────────┐          ┌─┴────────────┐
  │  5           │          │  5           │
  └──────────────┘          └──────────────┘
```

6		6
+		+
11		12

General information	Pure Data	Excel	LabVIEW	Reaktor	VEE	Blender
Main reference	[P$^+$15]		[Nat01]	[Nat15]	[Agi11]	[Ble17]
Licensing	3-clause BSD	Proprietary	Proprietary	Proprietary	Proprietary	GNU GPL v2+
Initial release	1996	1985	1986	1999	1991	1995
Latest release	2016	2016	2016	2015	2013	2017
Application domain	Music	Office	Engineering	Music	Engineering	3D graphics

Design alternatives [Hil92]	Pure Data	Excel	LabVIEW	Reaktor	VEE	Blender
Box-line representation	Yes	No	Yes	Yes	Yes	Yes
Iteration	Yes (cycles)	Limited	Yes (construct)	Limited	Yes	No
Subprogram abstraction	Yes	No	Yes	Yes	Yes	Yes
Selector/distributor	Yes	Yes	Yes	Yes	Yes	Yes
Flow of data	Uni	Uni	Uni	Uni	Uni	Uni
Sequence construct	No	No	Yes	No	Yes	No
Type checking	Limited	No	Yes	Yes	No	Yes
Higher-order functions	No	No	No	No	No	No
Execution mode	Data-driven	Demand-driven	Data-driven	Demand-driven	Data-driven	Data-driven
Liveness level [Tan90]	2	3	2	2	2	3

Additional design alternatives	Pure Data	Excel	LabVIEW	Reaktor	VEE	Blender
Dataflow model	Dynamic	Static	Static	Static	Static	Static
N-to-1 inputs	Yes	No	No	No	No	No
Separate edit/use views	No	No	Yes	Yes	Yes	No
Time-dependent firing	Yes	No	Yes	Yes	Yes	No
Rate-based evaluation	Synchronous	No	No	Synchronous	No	No
Indirect connections	Yes	Yes	Yes	Yes	Yes	No
Dynamic connections	Yes	Yes	Yes	No	No	No
Textual sub-language	Imperative	Functional	Imperative	No	Imperative	No
Scripting	Python, Lua	VBA	MATLAB	Reaktor Core	MATLAB	OSL, Python

Table 8.1: A comparison of contemporary dataflow UI-level languages

In short:

- scripting is not the only way to enable end-user programming,
  UI-level programmability works wonders

In short:

- scripting is not the only way to enable end-user programming, UI-level programmability works wonders

- but needs to be approached as *programming language design*

In short:

- scripting is not the only way to enable end-user programming, UI-level programmability works wonders

- but needs to be approached as *programming language design*

- many successful applications use *dataflow* languages and there is a *wealth of research* in the area

http://hisham.hm/thesis