# End-User Programming

How do casual users produce software?

The Problem

# What is most software like today?

From "I can't figure out how to..."

**Like data, menu options are proliferating.**

'Products such as Lotus Notes, Adobe Photoshop, Intuit Quicken, and Microsoft Word are so encrusted with a bewildering array of features that users are confounded and use few of them effectively, if at all.' - Alan Cooper

**User needs evolve faster than the development cycle.**

The 'latest release' product-deployment model doesn't keep up
Localization by labels does not meet expectations

**Even when the right tool is out there, somewhere, it can't be found**

No common language for procedures
Too many results
No credibility

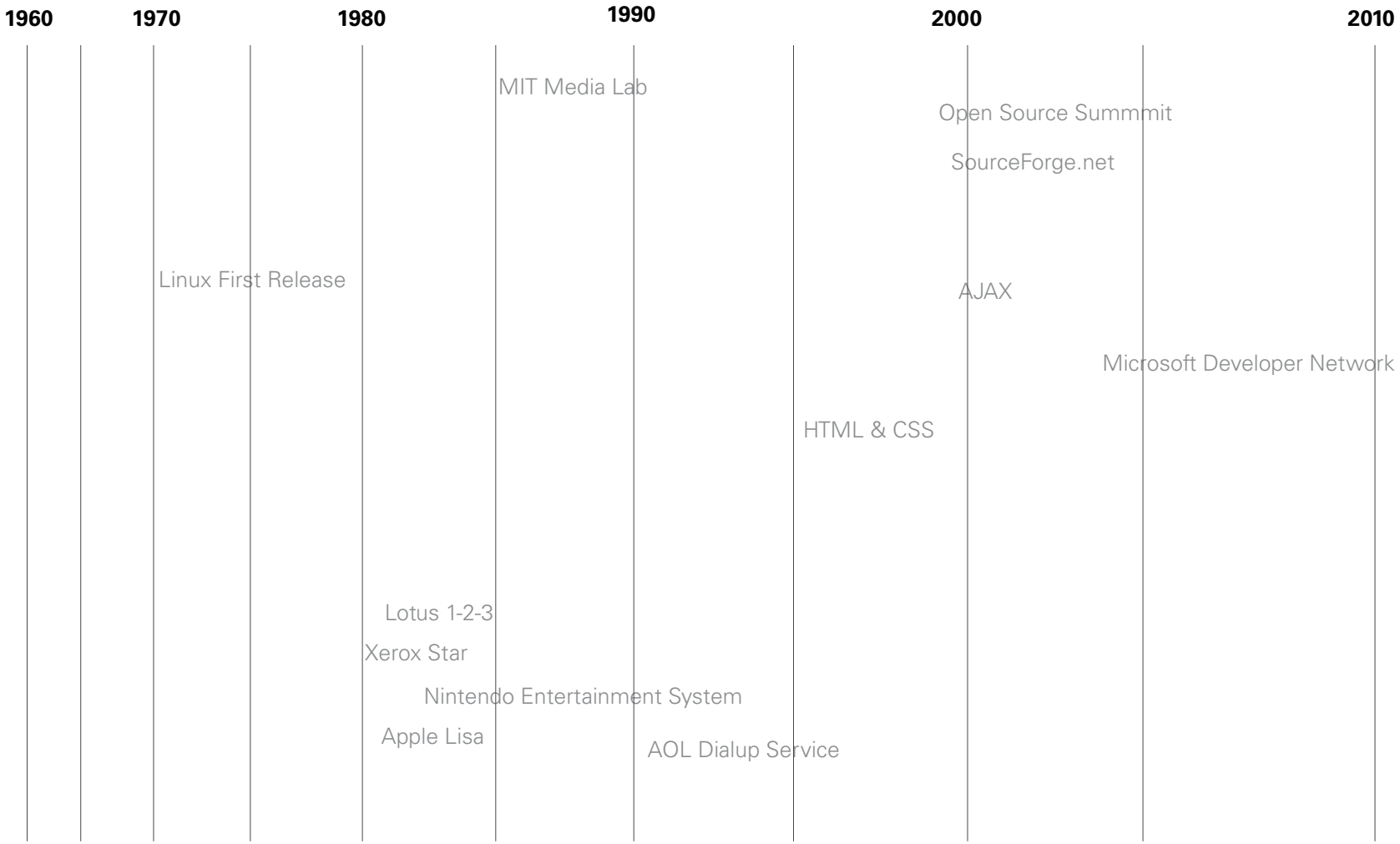
To "One way or another, I know I can get there..."

**Enabling the User**

To create applications
To add to others' applications
Without specialized training

History of end-user interfaces

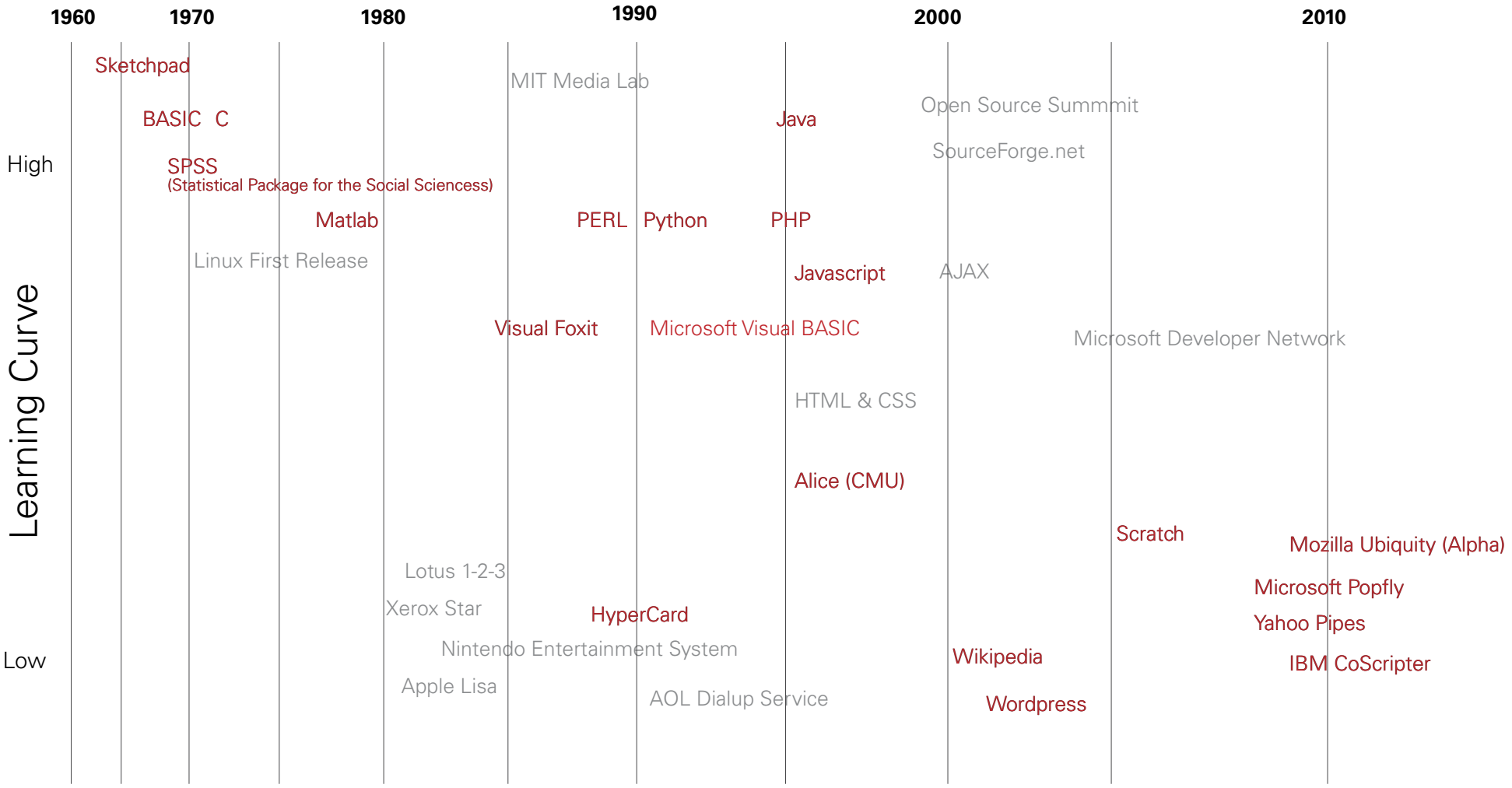
# From engineering to designing




## Background

History of end-user interfaces

# From engineering to designing

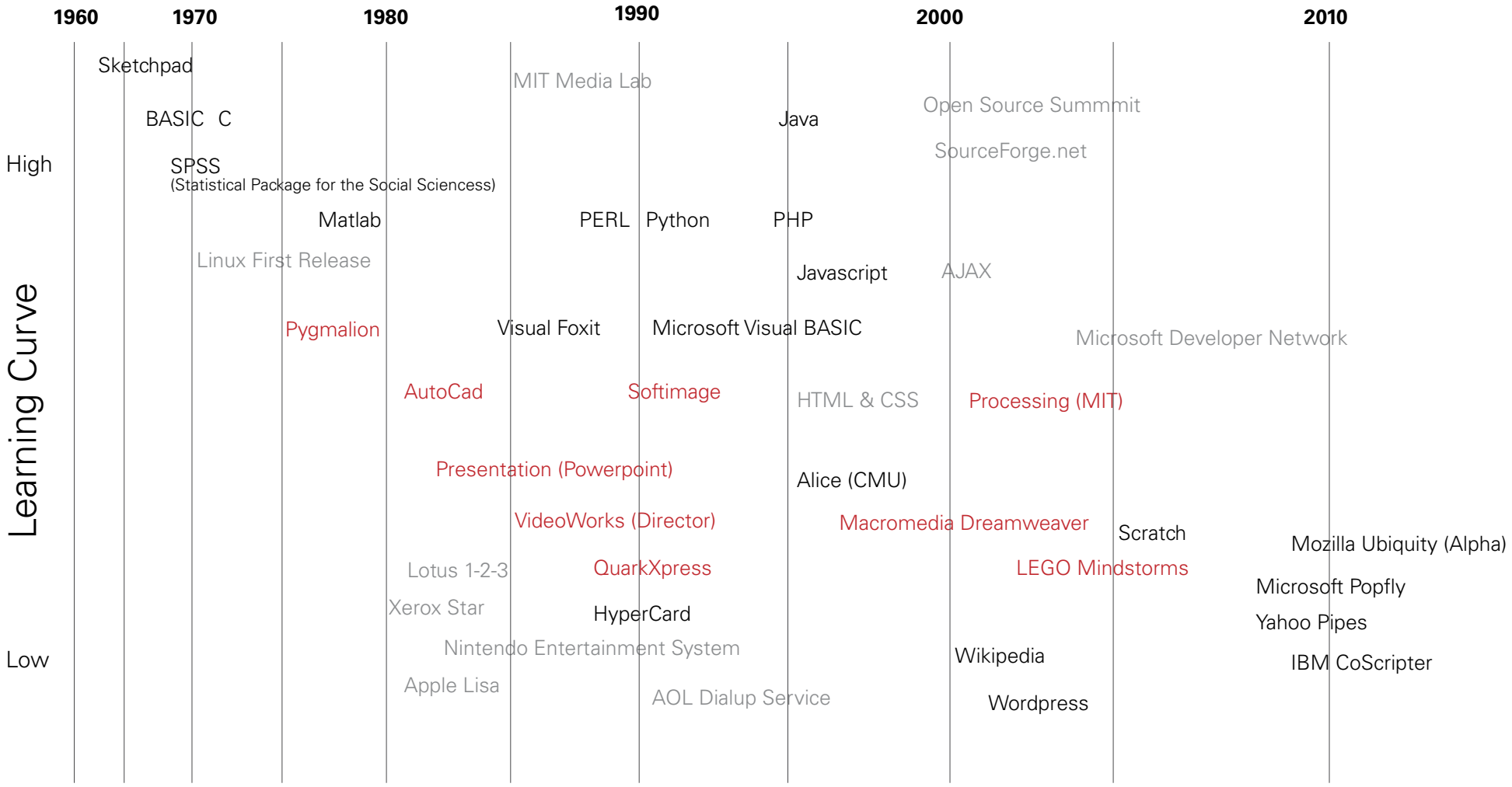



Background

General Purpose Programs

History of end-user interfaces

# From engineering to designing



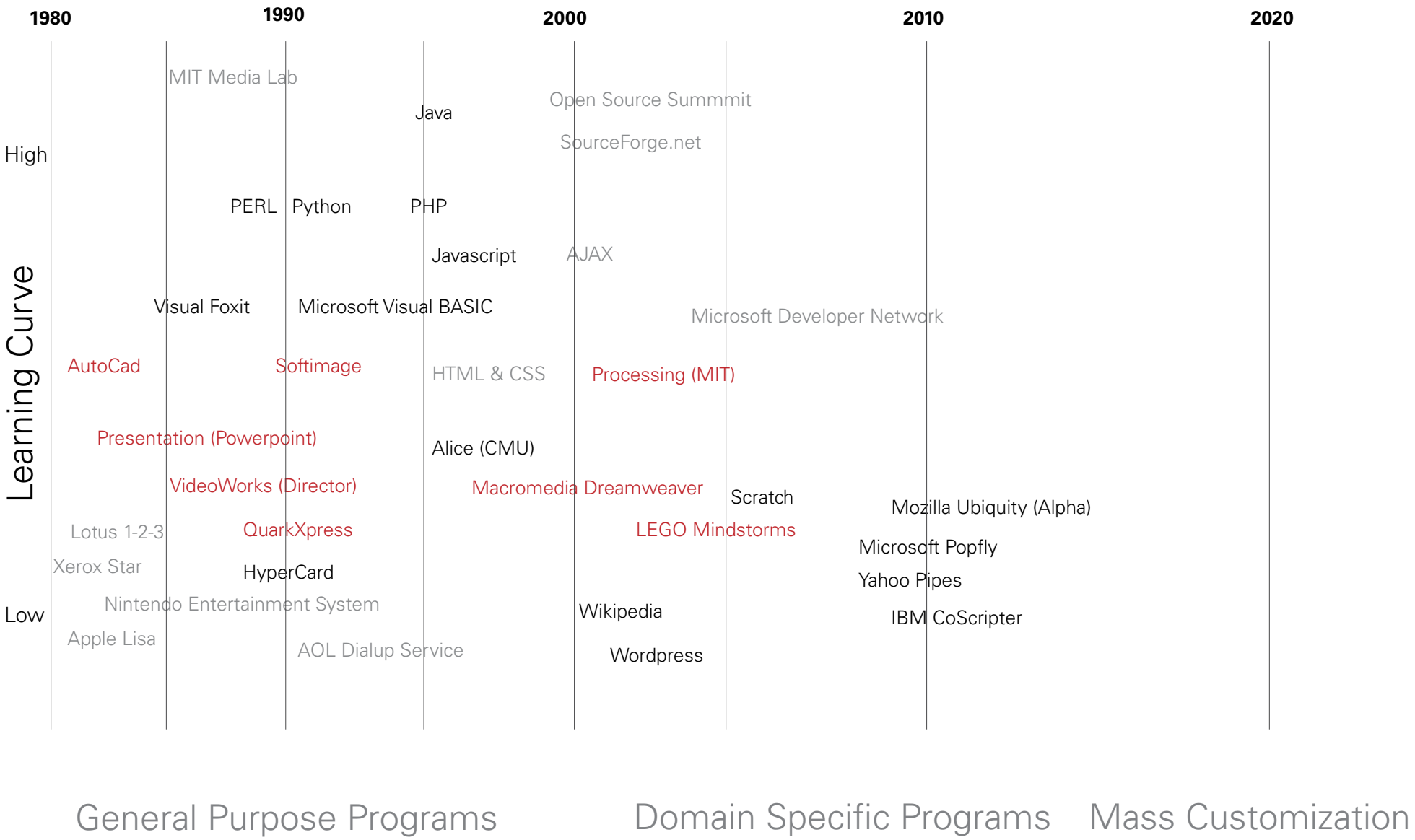

Background

General Purpose Programs

Domain Specific Programs

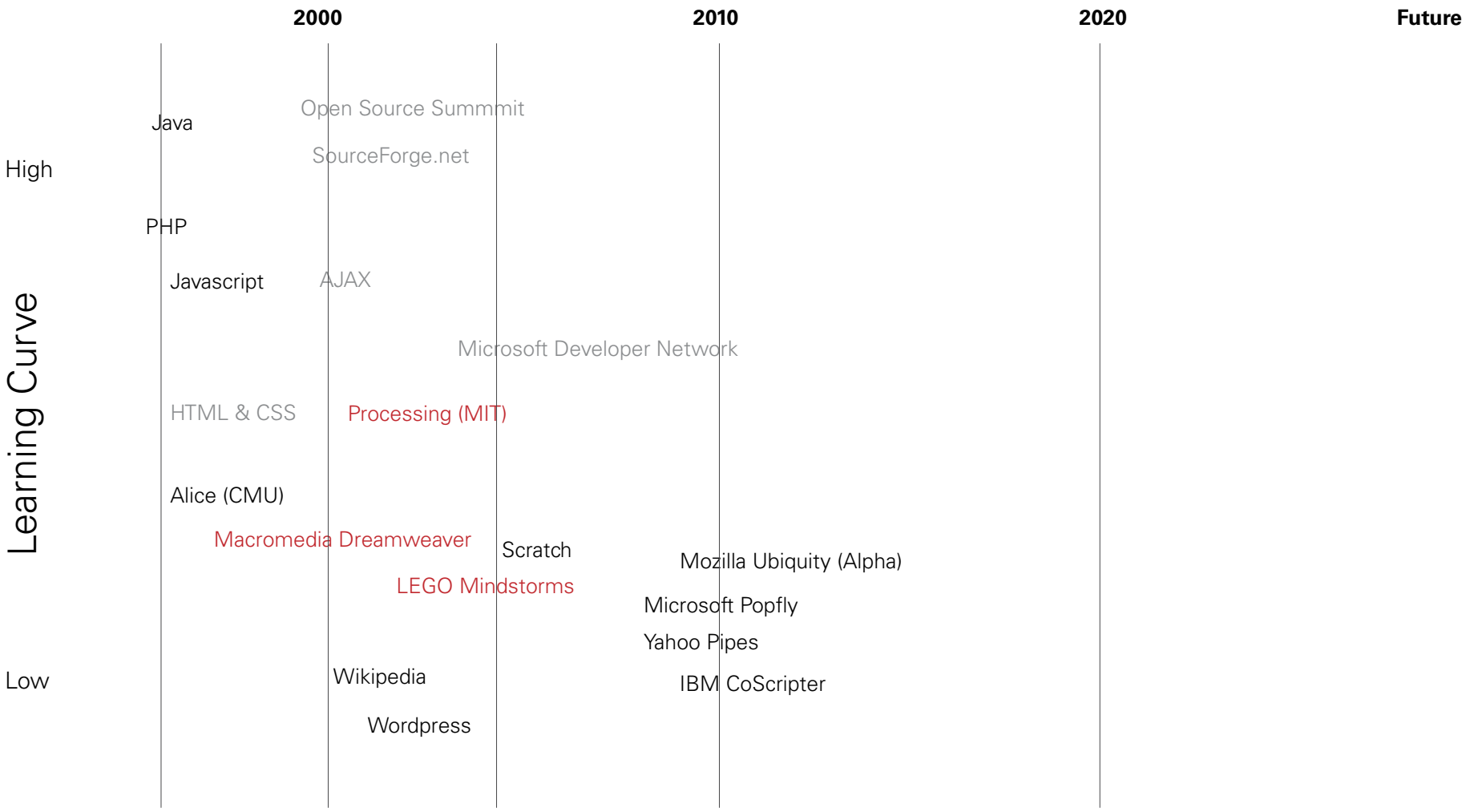
History of end-user interfaces

# From engineering to designing



History of end-user interfaces

# From engineering to designing




rpose Programs

Domain Specific Programs

Mass Customization

Personal Programming Languages

Current State of Affairs:
# Choose your adventure

**“Most users are neither beginners nor experts; instead, they are intermediates.”**

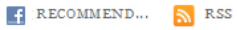
- Alan Cooper

## End-User Interaction Taxonomy


### High-level commands

Best Practices in interface design

Usually domain-specific




### Add-Ons

Users choose functions from a vast library

Usually open-source

### Macro Recorders

Users program by example

Playback is repeatable


### Integrated Development Environments

Reduces cognitive load on programmer

Often includes GUI builder


### Visual programming languages

Visual modelling of logic and flow

Either network diagrams or flow-charts

Examples:

# High-level commands

Basically, these are the current gems of functional interfaces.
Other examples:

## Wordpress

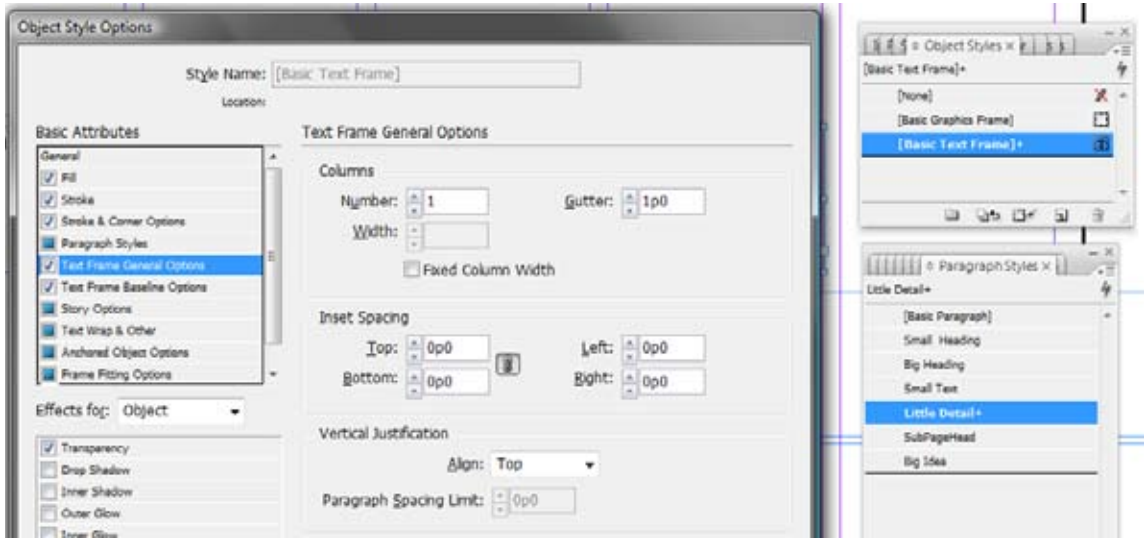
Leverages MS Word interface

Provides templates


## InDesign Styles

Object oriented

Based on set theory

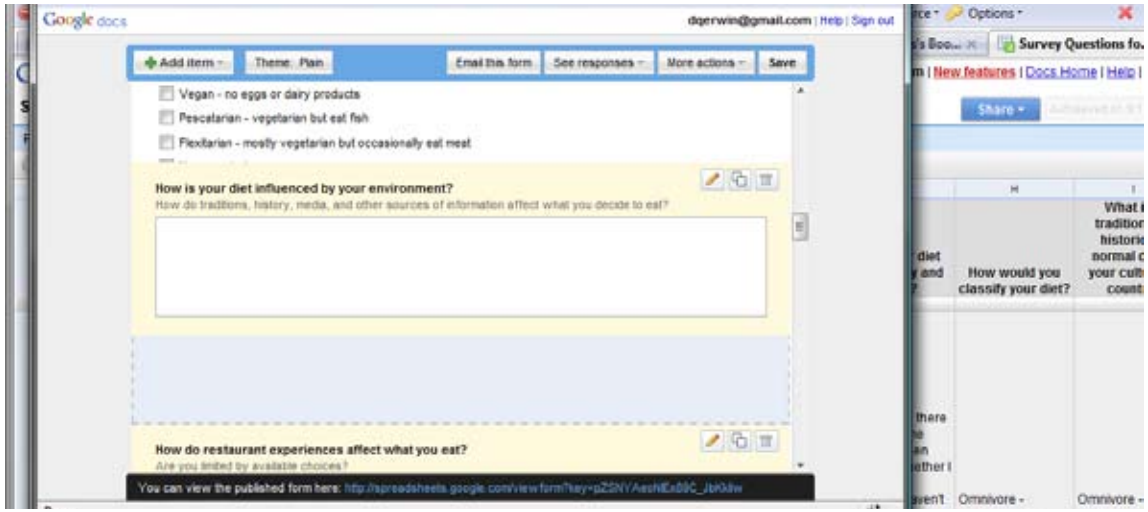



## Google Forms

Build Surveys

Drag & Drop

Zero coding

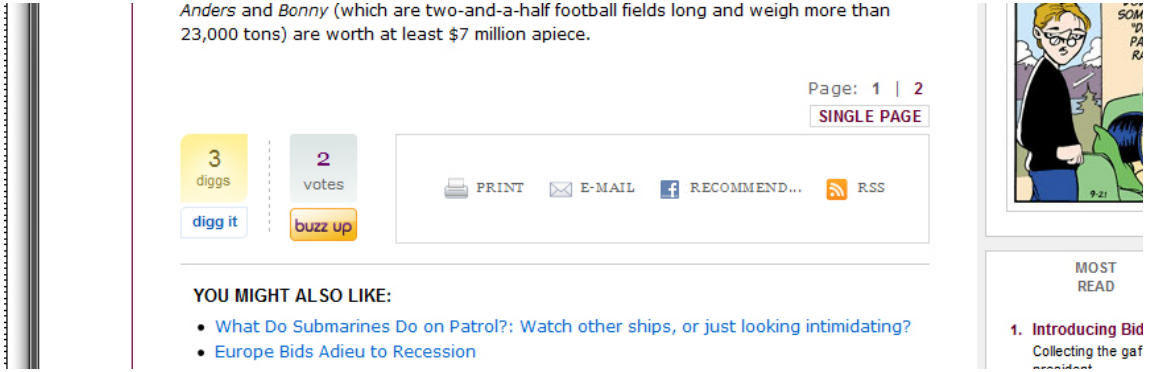


Examples:

# Add-Ons

## RSS Feeds

- One click on a web page sets up a recurring API interface

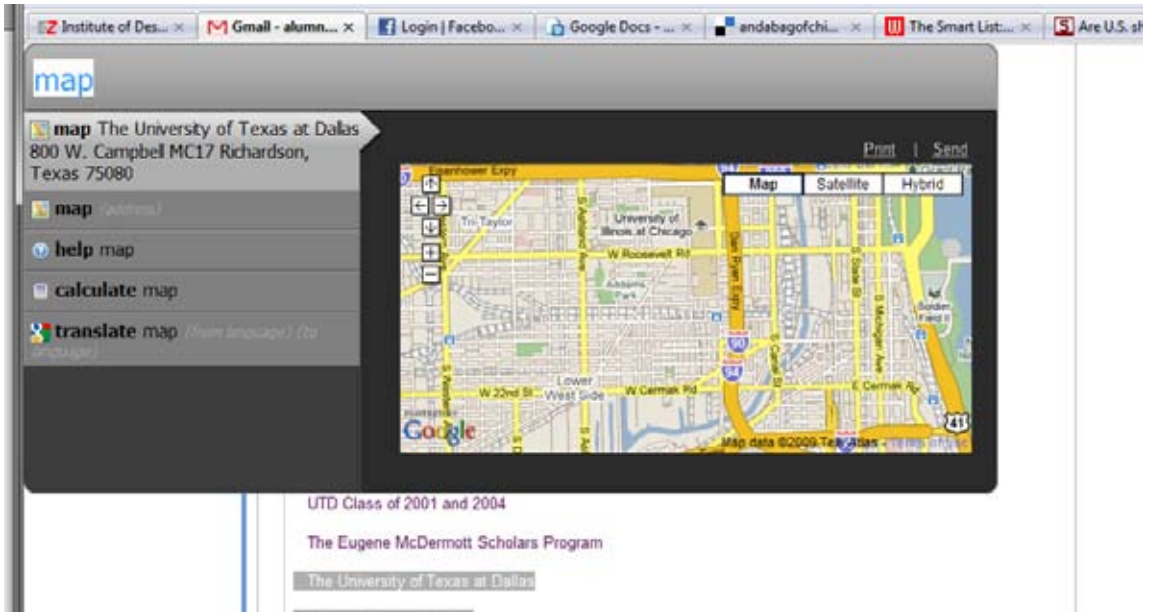



## Firefox Add-ons

- One click adds new browser tools
- Choose from thousands of free options


## Mozilla Labs: Ubiquity

- Command line
- Auto-updates new script versions

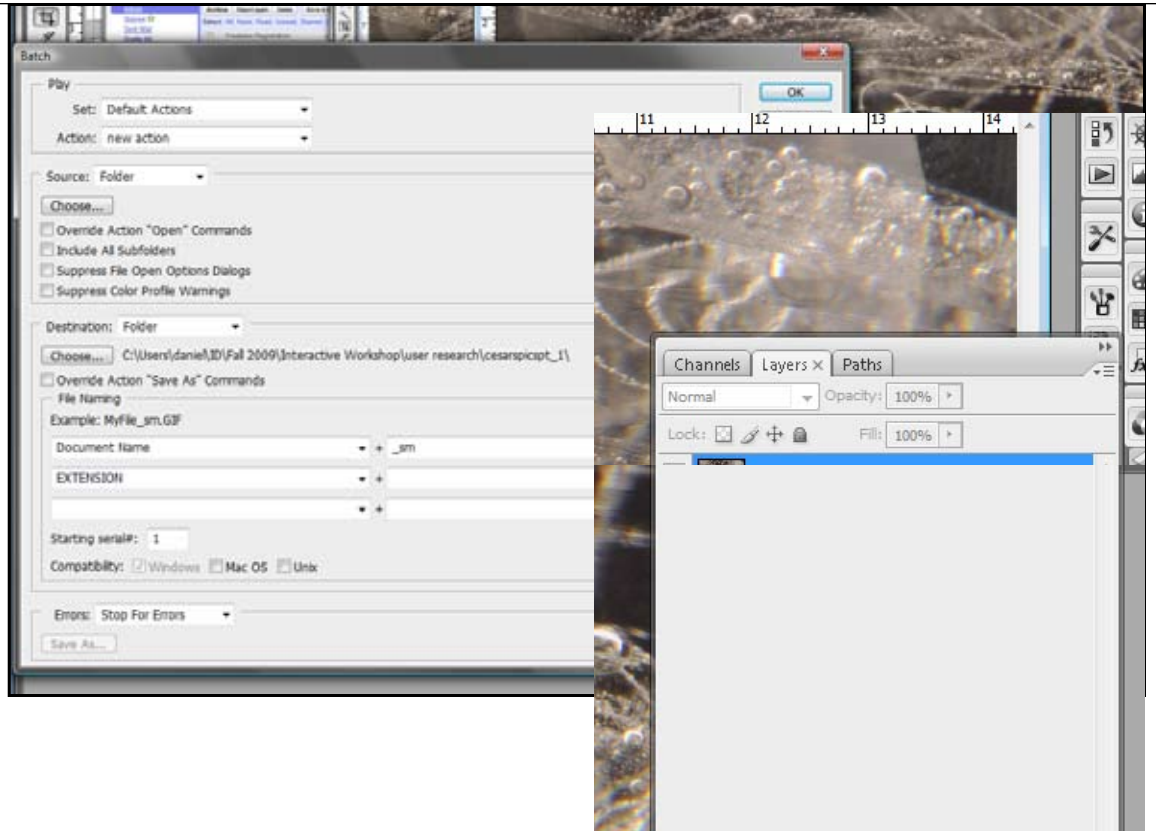


Examples:
# Macros

## Photoshop

- With the ‘record’ button click (as shown right), perform an operation.

- Later, automatically do the same to any group of images through the ‘batch’ window.



Examples:

# Integrated Development Environments

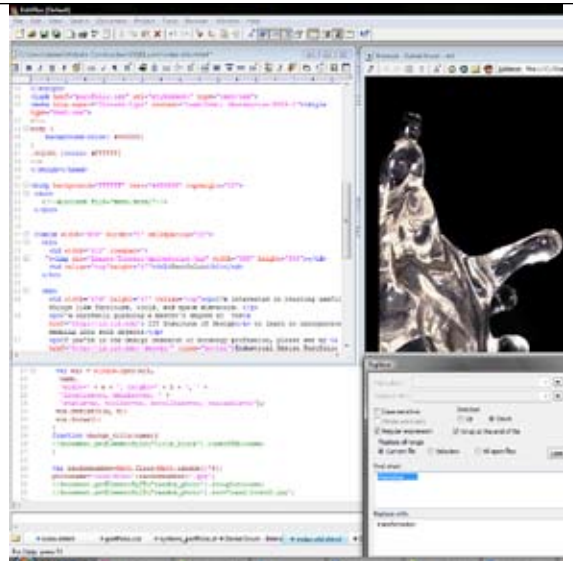
These tools make some parts of application publishing available to casual users but only templates for behaviors and processing.

## Edit Plus

- Color coding of language types helps scan code quickly

- Distinct color systems for each language (i.e. HTML vs JavaScript vs PHP) helps user intuit context.

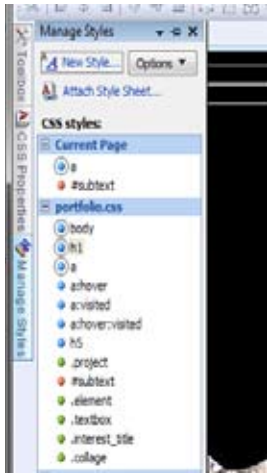
- Split-screen view

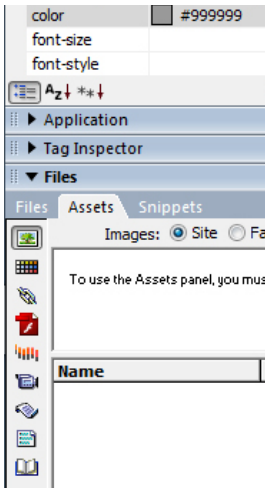


## Adobe Dreamweaver

- Properties window clarifies available actions

- Graphic type representations facilitate unconscious associations



Examples:

# Visual Programming Languages

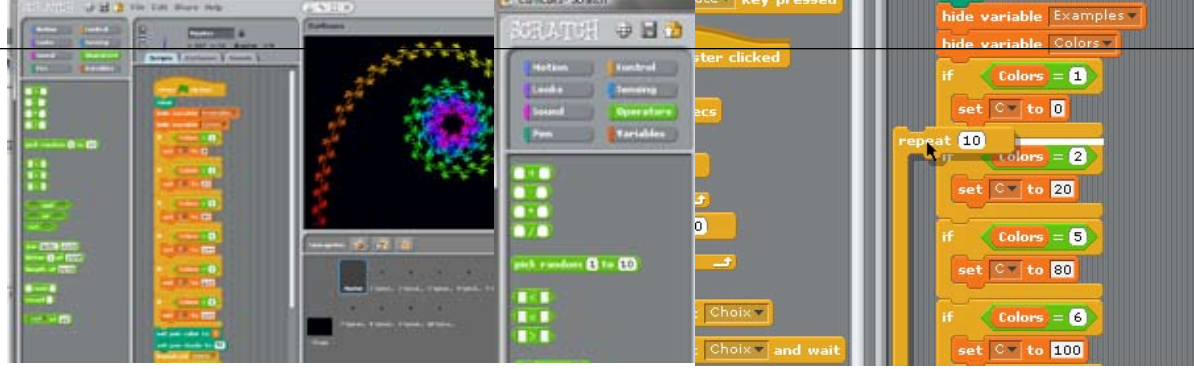
Historically the most active field of research in End-user programming, visualizations are making a comeback through UML-to-code projects.

## Scratch

Constraints represented by puzzle pieces

Language types grouped by color

Pieces snap into place

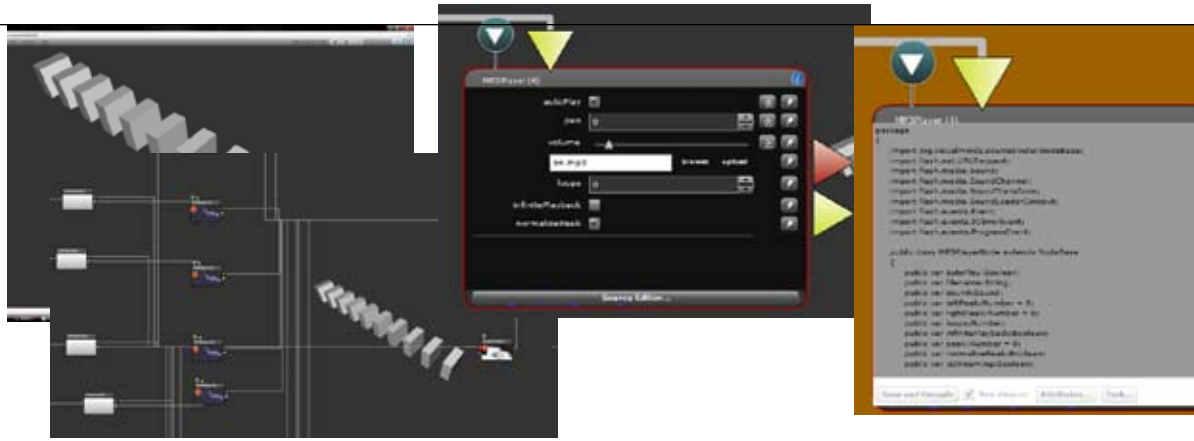



## Source Binder (alpha release)

Code modules layered on Flash Double-clicking opens module for editing

Clicking again reveals AS3 code

Includes libraries for 3d-modeling, APIs, and more

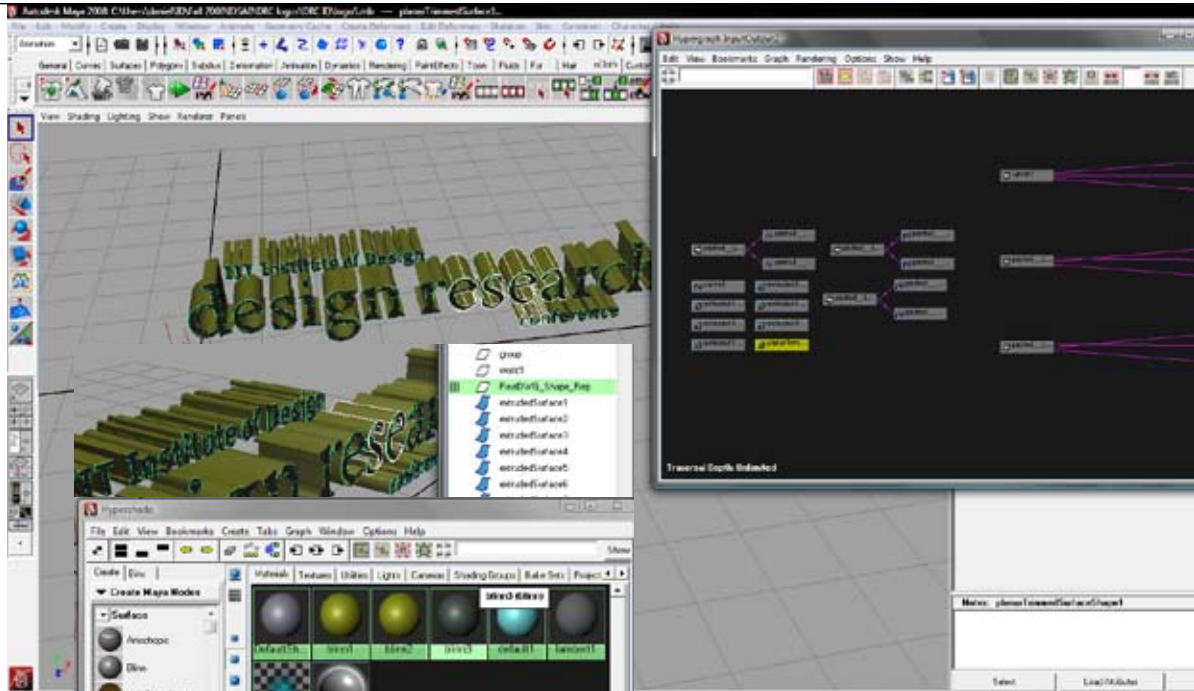


## Autodesk Maya

Anything can be linked to anything

Attributes of any object can be modified

Multiple editors keep track of different levels of relations

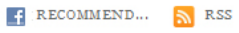


Future Predictions:
# Create your adventure

**Like finger paint (and unlike television), computers can be used for designing and creating things.**

- Mitch Resnick creator of Scratch

## Exampes




## Current Trends

**High-level commands**
Proliferation of commands
Enterprise customization

**Add-Ons**
Individuals and companies increasingly share code
Sharing data and code is easier with standards

**Macro Recorders**
Machine Learning algorithms
Diffusion of Computer Science concepts

**Integrated Development Environments**
UML-to-code tools
Black box connections to networks and APIs

**Visual programming languages**
Popularity of graphics software
Switching cost of new tools

## What's Next?

**Extensible Software**
Operating System providers and major development houses will support 'hacking' and mixing of their products with third- and fourth party software.

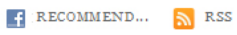
*Viruses.*

Future Predictions:
# Create your adventure

**Like finger paint (and unlike television), computers can be used for designing and creating things.**

- Mitch Resnick creator of Scratch

## Exampes




## Current Trends

**High-level commands**
Proliferation of commands
Enterprise customization

**Add-Ons**
Individuals and companies increasingly share code
Sharing data and code is easier with standards

**Macro Recorders**
Machine Learning algorithms
Diffusion of Computer Science concepts

**Integrated Development Environments**
UML-to-code tools
Black box connections to networks and APIs

**Visual programming languages**
Popularity of graphics software
Switching cost of new tools

## What's Next?

**AI Learning**
Users will take a few minutes to demonstrate an abstract process to the computer and test the computer's comprehension on test cases in several contexts. Visual representations of the learned behavior will support easy debugging and remixing.

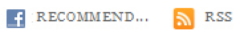
*Without deep knowledge of all involved software, it will not be obvious what a computer might or might not be able to learn.*

Future Predictions:
# Create your adventure

**Like finger paint (and unlike television), computers can be used for designing and creating things.**

- Mitch Resnick creator of Scratch

## Exampes




## Current Trends

### High-level commands

Proliferation of commands

Enterprise customization

### Add-Ons

Individuals and companies increasingly share code

Sharing data and code is easier with standards

### Macro Recorders

Machine Learning algorithms

Diffusion of Computer Science concepts

### Integrated Development Environments

UML-to-code tools

Black box connections to networks and APIs

### Visual programming languages

Popularity of graphics software
Switching cost of new tools

## What's Next?

### Visual Programming

Many object and processes will be manipulated with intuitive visual diagrams, allowing users to build programs to meet their needs exactly.

*There will be many processes that are difficult or impossible without traditional coding or even calculation-heavy math.*

## General Overviews

Fischer, G., Nakakoji, K., Ye,Y. Metadesign: Guidelines for Supporting Domain Experts in Software Development. IEEE Software September/October 2009.

Resnick, M. (2002). Rethinking Learning in the Digital Age (http://web.media.mit.edu/~mres/)

## Visual Programming Languages

Boshernitsan, M., Downes, M. (2004). Visual Programming Languages: A Survey. Report No. UCB/CSD-04-1368, Computer Science Division, EECS, University of California, Berkeley.

David Harel and Gordon-Kiwkowitz, M. (2009). On Teaching Visual Formalisms.

May/June 2009 IEEE Software.

Topçu, O., Adak, M., O˘guztüzün,H. (2009). Metamodeling live sequence charts for code generation. Software and Systems Modeling , Volume 8, Number 4 / September, 2009.

## End-users as Designers

Winograd, T. (1995). From Programming Environments to Environments for Designing. Communications of the ACM. Volume 38 , Issue 6.

Resnick, M., and Silverman, B. (2005). Some Reflections on Designing Construction Kits for Kids. Proceedings of Interaction Design and Children conference, Boulder, CO. (http://info.scratch.mit.edu/Research)

## Examples

CoScripter: http://coscripter.research.ibm.com/coscripter/browse/

StructureSynth

SourceBinder: http://sourcebinder.org/

Maya: http://autodesk.com/

Alice: http://www.alice.org/

Mindstorms NXT: http://www.ni.com/academic/mindstorms/

Microsoft Robotics: http://msdn.microsoft.com/en-us/robotics/default.aspx

Scratch: scratch.mit.edu/

## Quote

Cooper, A. (1999). The Inmates are Running the Asylum, p. 244.