

# The Changing (?) Face of Programming

**Brian W Kernighan**

Department of Computer Science  
Princeton University

## A new programming world?

"Mr. Hong describes himself as well off enough so that work is optional. He was collecting more than \$1 million a year from HotOrNot, a project he and his partner had created in seven days ..."

New York Times, 10/28/07

languages

tools

interfaces

components

methodologies

"Language shapes the way  
we think and determines  
what we can think about."

Benjamin Whorf (1897-1941),  
*Language, Thought and Reality*



**"What languages do you focus on?"**

parent of a prospective undergrad

"C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do it blows your whole leg off."

Bjarne Stroustrup, creator of C++



# Scripting languages

- the nature of many applications has changed
  - gluing components, web page creation
- scripting languages (Perl, Python, PHP, Ruby, ...)
  - avoid C & C++ memory management, awkward string handling, ...
  - more expressive than Java: e.g., RE's, string operators
  - casual type checking, object hierarchies, data structures
- comparatively weak tools
- slower but no one cares
  - computers are fast and cheap
  - programmers are slow and expensive

# Javascript

- very weak typing
- potentially risky object model
- inconsistent implementations, poor tools
- complicated irregular interface to browser
  - no life elsewhere
- no academic credentials

But:

- very widely used in many innovative systems
  - AJAX applications, libraries, web-based programs
- illustrates important programming ideas
  - event-driven programming, callbacks, closures
  - DOM: object oriented interface with a vengeance
  - JSON: a lightweight alternative to XML for structured data



Start address e.g., "SFO"

princeton, nj

End address e.g., "94526"

carnegie hall

Get Directions

Search the map

Find businesses

Get directions


# AJAX in action



Dept./Program - Office of Registrar - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://registrar1.princeton.edu/course/upcome/dept/crs1.cfm



# Princeton University

## Office of the Registrar

101 West College Hall  
Princeton, NJ 08542 - 0070  
Tel: (609) 258-3360  
Fax: (609) 258-6328

Academic Year Hours: 8:45a.m.-5:00p.m.  
Summer Hours: 8:30a.m.-4:30p.m.

### DEPARTMENT OR PROGRAM

Spring 2007 - 2008  
Last Updated: 01/04/08 4:00 pm

- [PU Home](#)
- [PU Search](#)
- [Registrar Home](#)
- [Academic Calendars](#)
- [Blackboard](#)
- [Classroom Information](#)
- [Course Information](#)
- [Data and Information](#)
- [Enrollment Verification and Transcript](#)
- [Final Exams](#)
- [General Academics](#)
- [Grading Policies](#)

To obtain a listing of the departments that start with the letter A, click on the letter A from the drop down list above and click on the letter A box to go to the list of departments that start with the letter A.

e.g. Select ANT-ANTHROPOLOGY for the present semester.

- AAS - Program in African-American Studies
- CLA - Classics
- CLG - Classical Greek
- COM - Comparative Literature
- COS - Computer Science**
- CWR - Program in Creative Writing
- CZE - Czech
- DAN - Dance
- EAP - Program in East Asian Studies
- EAS - East Asian Studies
- ECO - Economics
- ECS - Program in European Cultural Studies
- EEB - Ecology and Evolutionary Biology
- EGR - Engineering
- ELE - Electrical Engineering
- ENG - English
- FNV - Program in Environmental Studies

03 Errors Adblock

(This does not work with IE6, which appears not to understand standards. Use Firefox or Safari.)

start typing here:  (e.g., eng for English courses or qr mw for Monday/Wednesday QR courses or smith; mouse over results for details.)

COS 116	ST	TTh	01:30-02:50	COMPU 105	The Computational Universe
COS 126	QR	TTh	11:00-11:50	FRIST 302	General Computer Science
COS 217	QR	MW	10:00-10:50	COMPU 105	Introduction to Programming Systems
COS 226	QR	MW	11:00-12:20	FRIST 302	Algorithms and Data Structures
COS 320		MWF	10:00-10:50	FRIEN 008	Compiling Techniques
COS 333		TTh	11:00-12:20	COMPU 105	Advanced Programming Techniques
COS 398		-		Junior In	
COS 423		MW	11:00-12:20	FRIEN 008	This is a course about the practice of programming. Programming is more than just writing code. Programmers must also assess tradeoffs, choose among design alternatives, debug and test, improve performance, and maintain software written by themselves & others. At the same time, they must be concerned with compatibility, robustness, and reliability, while meeting specifications. Students will have the opportunity to develop these skills by working on their own code and in group projects. <i>Brian W. Kernighan 41946</i>
COS 424		TTh	11:00-12:20	COMPU 104	
COS 426		TTh	01:30-02:50	EQUAA A224	
COS 435		MW	01:30-02:50		
COS 444	SA	TTh	01:30-02:50	COMPU 102	
COS 461		MW	01:30-02:50	COMPU 105	
COS 498		-		Senior In	
COS 511		MW	01:30-02:50	FRIEN 008	Theoretical Machine Learning
COS 598A		F	01:30-04:20	COMPU 402	Advanced Topics in Computer Science: Parallel Architec
COS 598B		M	01:30-04:20	COMPU 301	Advanced Topics in Computer Science: Vision: From Neur
COS 598C		TTh	01:30-02:50	COMPU 301	Advanced Topics in Computer Science: Geometric Modelin
COS 598D		F	01:30-04:20	COMPU 301	Advanced Topics in Computer Science: Mathematical Meth
COS 598F		F	01:30-04:20	COMPU 401	Advanced Topics in Computer Science: Biological Networ

[Students](#) | [Prospective Students](#) | [Parents](#) | [Alumni](#) | [Faculty & Staff](#)[About](#) | [Academics](#) | [Library](#) | [Research](#) | [Admission & Aid](#) | [Administration & Services](#) | [Campus Life](#) | [Visiting C](#)[A to Z Lists](#)

Search A to Z

[Submit](#)

Filter by Letter

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#)[H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#)[O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#)[V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)**Browse All Websites: A**[Academic Calendar](#)[Academic Managers Group \(AMG\)](#)[Accounting, Office of Sponsored Research](#)[Accounts Payable and Travel Accounting, Office of](#)[Administration, Vice President for \(see: Executive Vice President,](#)[Office of the\)](#)[Administrative Information Services, Office of Information Technology](#)[Admission, Graduate School](#)[Admission, Undergraduate](#)[African American Studies, Center for](#)

# All web site search

Start typing here:

[Administrative Information Services, Office of Information Technology](#)

[Computing \(Office of Information Technology\)](#)

[Information Systems, Administrative](#)

[Information Technology, Office of](#)

[Mailing, Office of Printing and](#)

[Office of Information Technology \(OIT\)](#)

[OIT Solutions Center](#)

[Printing and Mailing, Office of](#)

[Solutions Center, OIT](#)

[Vice President for Information Technology & CIO, Office of the](#)

[Web Support, Office of Information Technology](#)

**"A language that doesn't affect the way you think about programming is not worth knowing."**

*Alan Perlis, Epigrams on Programming*



languages

**tools**

interfaces

components

methodologies

**"Give us the tools and  
we will finish the job."**

Winston Churchill,  
radio address,  
February 9, 1941



# Tools

- mechanization of repetitive / tedious / error-prone tasks
- Unix introduced the conscious focus on creating and using tools
  - small tools: `grep`, `diff`, `wc`, `sort`, `uniq`, ...
  - specialized tools: `yacc`, `lex`, `make`
  - programmable tools: `shell`, `awk`
- Windows popularized IDEs (integrated development environments)
  - complete systems for editing, compiling, debugging, packaging

# Visual Studio

The screenshot displays the Microsoft Visual Studio IDE. The main window shows a code editor with the following C# code snippet:

```
protected void OnNewFeed(object sender, ServiceEventArgs e)
{
    Tracing.TraceMsg("Created new feed");
    if (e == null)
    {
        throw new ArgumentNullException("e");
    }
    if (e.Uri.AbsoluteUri.IndexOf("/") +
        GDataSpreadsheetsNameTable.FeedCell + "/" != -1)
    {
        e.Feed = new CellFeed(e.Uri, e.Service);
    }
    else if (e.Uri.AbsoluteUri.IndexOf("/") +
        GDataSpreadsheetsNameTable.FeedList + "/" != -1)
    {
        e.Feed = new ListFeed(e.Uri, e.Service);
    }
}
```

The Class View on the right shows the project structure, including the 'Spreadsheets' folder and 'SpreadsheetsSampleApp'. The Error List at the bottom shows 20 errors, 0 warnings, and 0 messages. The errors are summarized in the table below:

Line	Description	File	Line	Column	Project
1	The type or namespace name 'Spreadsheets' does not exist in the namespace 'Google.GData' (are you missing an assembly reference?)	appsdemo.cs	7	20	gapps_sample
2	The type or namespace name 'SpreadsheetsService' could not be found (are you missing a using directive or an assembly reference?)	appsdemo.cs	29	48	gapps_sample
3	The type or namespace name 'SpreadsheetEntry' could not be found (are you missing a using directive or an assembly reference?)	appsdemo.cs	30	13	gapps_sample
4	The type or namespace name 'SpreadsheetsService' could not be found (are you missing a using directive or an assembly reference?)	appsdemo.cs	51	63	gapps_sample
5	The type or namespace name 'SpreadsheetsService' could not be found (are you missing a using directive or an assembly reference?)	appsdemo.cs	71	46	gapps_sample
6	The type or namespace name 'WorksheetEntry' could not be found (are you missing a using directive or an assembly reference?)	appsdemo.cs	71	75	gapps_sample
7	The type or namespace name 'SpreadsheetsService' could not be found (are you missing a using directive or an assembly reference?)	appsdemo.cs	108	45	gapps_sample

# Eclipse

The screenshot displays the Eclipse IDE interface with the following components:

- Package Explorer:** Shows the project structure for `com.aramco.powers2.ui`, including source files and a test directory containing `NbBundleTest.java`.
- Editor:** Displays the source code of `NbBundleTest.java`. The code includes imports, annotations, and two test methods: `testExistingResource()` and `testNonExistingResource()`. A `main` method is also present for direct execution.
- Outline:** Shows the class structure of `NbBundleTest` with its methods: `main(String[])`, `testExistingResource()`, and `testNonExistingResource()`.
- Console:** Shows the execution output, indicating that the test finished after 0.129 seconds with 2/2 runs, 0 errors, and 0 failures.
- JUnit View:** Displays the test results for `com.aramco.powers2.ui.test.NbBundleTest` using the JUnit 4 runner.

```
27 import com.aramco.powers2.ui.NbBundle;
28
29 /**
30  * Tests the behavior of utility class NbBundle.
31  * Tests need to run against the background of a known set of objects.
32  * This set of objects is called a test fixture. (Refer to http://www.junit.org)
33  */
34  * @author Guanglin Du (dugl@petrochina.com.cn), Software Engineering Center, RIPED, PetroChina
35  */
36 public class NbBundleTest {
37
38     /**
39      * Uses the Bundle.properties to test NbBundle's behavior.
40      */
41     @Test
42     public void testExistingResource() {
43         String s1 = NbBundle.getMessage(ProjectView.class, "add_new_pvt_sat");
44         assertEquals("Add New PVT or SAT table", s1);
45     }
46
47     /**
48      * Uses the Bundle.properties to test NbBundle's behavior.
49      */
50     @Test
51     public void testNonExistingResource() {
52         String s1 = NbBundle.getMessage(ProjectView.class, "non-existing");
53         assertEquals("%non-existing", s1);
54     }
55
56     /**
57      * Method main to run this class directly.
58      * Can be run this way also on a command line:
59      * java org.junit.runner.JUnit4 samples.SimpleTestFixture
60      */
61     public static void main(String args[]) {
62         JUnitCore.main("com.aramco.powers2.ui.util.test.NbBundleTest");
63     }
64 }
65
```

Finished after 0.129 seconds

Runs: 2/2    Errors: 0    Failures: 0

com.aramco.powers2.ui.test.NbBundleTest [Runner: JUnit 4]

Failure Trace

Read Me Trim (Bottom)

# Web Frameworks

- software systems to create web applications
  - client (web) front end, server code, database access, administrative subsystem, etc.
- programs that write programs
  - generate code from specification plus conventions
- examples:
  - Rails (Ruby)
  - Django (Python)
  - Google Web Toolkit (Java)
  - and zillions of others [Wikipedia lists nearly 130!!]

# Google Web Toolkit

- write client (browser) code in Java
- including GUI layout, using classes like Java Swing
- test client code on server side
- once this works, compile Java to Javascript and HTML
- use the generated code on the client side

**Mortgage Calculator**

Update Clear Quit

Principal  
10000

Interest Rate  
10

Monthly Payment  
100

1	83.33	16.67	9983.33
2	83.19	16.81	9966.53
3	83.05	16.95	9949.58
4	82.91	17.09	9932.50
5	82.77	17.23	9915.27
6	82.63	17.37	9897.89
7	82.48	17.52	9880.38
8	82.34	17.66	9862.71
9	82.19	17.81	9844.90
10	82.04	17.96	9826.94
11	81.89	18.11	9808.83
12	81.74	18.26	9790.57
13	81.59	18.41	9772.16

**Mortgage Calculator**

Update Clear Quit

Principal  
10000

Interest Rate  
10

Monthly Payment  
100

1	83.33	16.67	9983.33
2	83.19	16.81	9966.53
3	83.05	16.95	9949.58
4	82.91	17.09	9932.50
5	82.77	17.23	9915.27
6	82.63	17.37	9897.89
7	82.48	17.52	9880.38
8	82.34	17.66	9862.71
9	82.19	17.81	9844.90
10	82.04	17.96	9826.94
11	81.89	18.11	9808.83
12	81.74	18.26	9790.57
13	81.59	18.41	9772.16

# Django: Python web framework



**Spare Stuff**

finding use for your spare goods.

[info](#) | [charities](#) | [mystuff](#)



**Search for charities that can utilize  
your spare goods or excess inventory**

Search bicycles, business attire, cars, laptops, music instruments, prom dresses, recliners, school supplies, tables, to

**Let your spare stuff be put to good use...**



# Spare Stuff

finding use for your spare goods.

[info](#)[charities](#)[account](#)[Charities](#)[By Popularity](#)[By Category](#)[Login](#)

## Top 10 Charities

[Top 10](#)[Top 25](#)[Top 50](#)

Charities are ranked by how many users click on their page for more information.

1

### Goodwill Industries Of Greater New York And Northern New Jersey, Inc.

**Astoria, NY:** Goodwill Industries is a leading provider of vocational, youth and community redevelopment services for people with disabilities and other special needs in New York and Northern New Jersey. Its mission is to expand opportunities and job capabilities for people outside the mainstream workforce, whether they are mentally or physically disabled, economically disadvantaged, recent immigrants, unskilled workers or urban youth. The agency operates its programs in the five boroughs of New York City and the metropolitan area, including Long Island and ... [\(more info\)](#)

2

### Goodwill Industries Of Southern New Jersey/Quaker City Goodwill

**Maple Shade, NJ:** Goodwill Industries of Southern New Jersey/Quaker City Goodwill is a nonprofit, community-based organization. Its mission is to provide education, job training and employment services that prepare individuals with disabilities and other disadvantaging conditions, such as welfare dependency, illiteracy, and spousal abuse, for competitive community employment. Revenue from the mass collection and resale of donated items in Goodwill's area retail stores fund employment training programs. [\(more info\)](#)



## Goodwill Industries Of Greater New York And Northern New Jersey, Inc.

4-21 27th Avenue, Astoria, NY, 11102

Goodwill Industries is a leading provider of vocational, youth and community redevelopment services for people with disabilities and other special needs in New York and Northern New Jersey. Its mission is to expand opportunities and job capabilities for people outside the mainstream workforce, whether they are mentally or physically disabled, economically disadvantaged, recent immigrants, unskilled workers or urban youth. The agency operates its programs in the five boroughs of New York City and the metropolitan area, including Long Island and Northern New Jersey. More info...



### Contact Information

Email: [info@goodwillny.org](mailto:info@goodwillny.org)

Phone: 718-728-5400

Website: <http://www.goodwillny.org> (preferred)

### Accepted Items

- clothing
- sporting goods
- housewares
- furniture

**"A most important ... aspect of any tool is its influence on the habits of those who train themselves in its use."**

*Edsger Dijkstra, A Discipline of Programming, 1976*



languages

tools

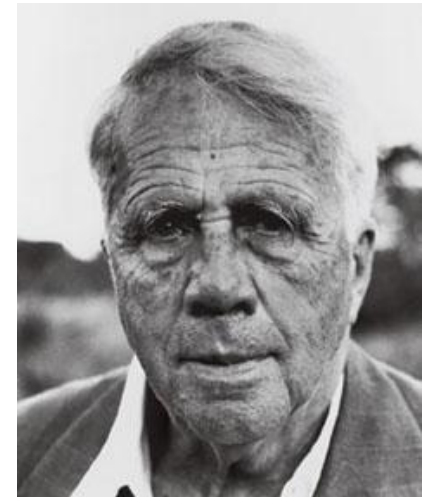
**interfaces**

components

methodologies

**"Good fences make good neighbors"**

Robert Frost,  
*Mending Wall*, 1914




# Interfaces

- the boundary between a service and its users
- "Every module ... is characterized by its knowledge of a design decision which it hides from all others. Its interface was chosen to reveal as little as possible about its inner workings."
  - David Parnas, "On the criteria to be used in decomposing systems into modules", *CACM*, 1972
- opaque types [e.g. *C* standard I/O library]
- objects
- web interfaces

# Google Maps API

```
<body>
  <div id="map" style="width: 800px; height: 600px"></div>
  <script type="text/javascript">
    //
    var here = new GPoint(-74.6550671545715, 40.3464459982649);
    var map = new GMap(document.getElementById("map"));
    map.addControl(new GLargeMapControl());
    map.addControl(new GMapTypeControl());

    map.setMapType(G_HYBRID_TYPE);
    map.centerAndZoom(here, 0);
    map.openInfoWindow(map.getCenterLatLng(),
      document.createTextNode("You are here, more or less"));
    //]]&gt;
  &lt;/script&gt;
&lt;/body&gt;</pre></div>
```

A satellite map of a campus area. A white speech bubble with a black border is centered over a large green field. The text inside the bubble reads "You are here, more or less". To the right of the text is a small blue square icon with a white 'X'. The map shows several buildings, parking lots, and roads. A road labeled "Washington Rd" runs diagonally across the bottom right. Another road labeled "Ivy Ln" is visible on the right side. A road labeled "Wash" is visible at the top. The field in the center has some faint markings on it.

You are here, more or less



# APIs in the News (November 2007)

 Search

e.g. "ajax apis" or "open source"

## OpenSocial

Hot

### The web is better when it's social

The web is more interesting when you can build apps that easily interact with your friends and colleagues. But with the trend towards more social applications also comes a growing list of site-specific APIs that developers must learn.

OpenSocial provides a common set of APIs for social applications across multiple websites. With standard JavaScript and HTML, developers can create apps that access a social network's friends and update feeds.

### Many sites, one API

Common APIs mean you have less to learn to build for multiple websites.



"Little wonder then that we see so many poorly designed APIs: it is not reasonable to expect programmers to be good at something they have never been taught."

Michi Henning, "API: Design Matters",

*ACM Queue*, May 2007



languages

tools

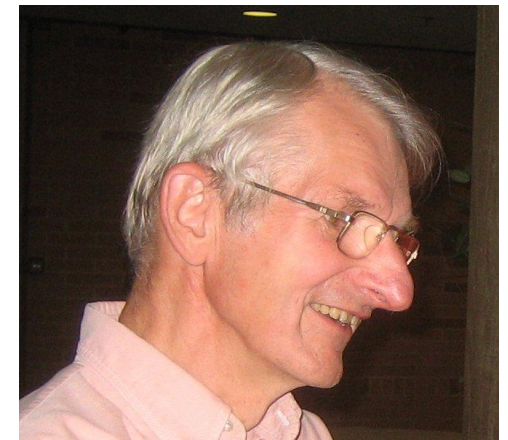
interfaces

**components**

methodologies

"We undoubtedly produce software by backward techniques. ... I would like to see components become a dignified branch of software engineering."

Doug McIlroy, *Mass Produced Software Components*, NATO Software Engineering Conference, 1968



# Pipes then and now

- one of earliest component implementations  
who | grep Joe | wc
- Yahoo pipes:

The screenshot displays the Yahoo Pipes web interface. On the left, a sidebar lists various operators under the 'Operators' category, including Count, Filter, Location Extractor, Loop, Regex, Rename, Reverse, Sort, Split, Sub-element, Tail, Truncate, Union, Unique, and Web Service. Below this, there are sections for 'Url' and 'String'. The main workspace shows a pipeline configuration on a grid background. A 'Yahoo! Search' module is connected to a 'Filter' module. The 'Filter' module is configured to 'Block' items that match 'all' of the following rules. One rule is selected: 'item.author' contains. A list of available fields for filtering is shown, including item.description, item.link, item.title, item.updatedon, and its sub-fields (day, day\_of\_week, hour, minute, month, second). A 'Pipe Output' module is connected to the end of the pipeline.

**Count**  
This module counts the number of items in a feed. The result is a typed number that you can pipe into a numeric input field within another module.

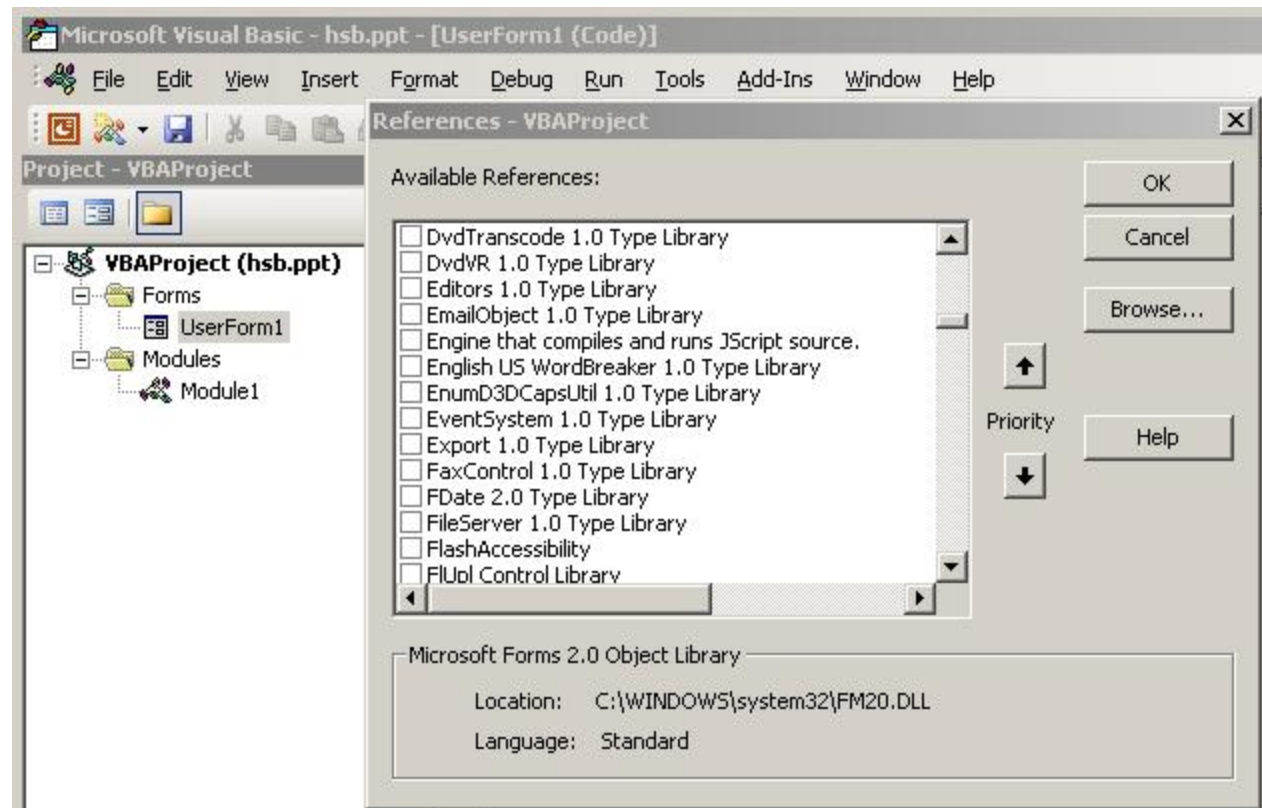
# Pipes versus type-checking

The screenshot shows the Pipes application interface. The top bar includes the 'pipes' logo, a tab labeled 'b1\*', and buttons for 'Layout', 'Expand All', and 'Collapse All'. A 'Back to My Pipes' link is visible in the top right. The left sidebar contains a tree view with categories: Sources, User inputs, Operators, Url, and String. The Operators list includes: Count, Filter, Location Extractor, Loop, Regex, Rename, Reverse, Sort, Split, Sub-element, Tail, Truncate, Union, Unique, and Web Service. Below the sidebar, the 'Count' module is detailed: 'Count' - This module counts the number of items in a feed. The result is a typed number that you can pipe into a numeric input field within another module. Example: [Using the Count Module](#). Learn more: [about this module](#).

The main workspace shows a workflow on a grid background. It consists of three modules connected by blue lines: a 'Yahoo! Search' module (top) with a search input field containing 'kernighan' and a 'Site restriction' option; a 'Count' module (middle) connected to the output of the search; and a 'Pipe Output' module (bottom) connected to the output of the count module.

# COM: the Component Object Model

- McIlroy's components: small libraries
- COM: objects on steroids
  - the most successful example of components in practice



# Mashups: duct tape programming

- the web version of components?
- the browser as operating system?

Home News **Mashups** APIs Markets How-To Contests Members


Overview Mashup Directory Most Popular Mashups **New Mashups by Date** Mashup Matrix Mashup Tag Cloud

## Keep Up With Jones

◀ 3 of 2473 Mashups by Date ▶

Summary Comments (0)

### Overview



**Address:**  
123 Main Street, SomeCity, ST, 12345

**Pricing Information:**  
Zestimate™: \$270,267.00  
Value Range: \$245,943.00 - \$297,294.00

**Value Range:**  
US \$ (1,000) \$0K \$270.27K \$297.29K

★★★★★ Click stars to vote

### Description

This Facebook app allows you to connect to Zillow from Facebook. View information about your house and see what your friends houses are worth.

**APIs** Facebook + Microsoft Virtual Earth + Zillow

**Tags** mapping, money, realstate, social, widgets

**Mashup of the Day** ✓

**Added** 02 Nov 2007

**Who** bbuffone [Profile]

**URL** [http://apps.facebook.com/keep ...](http://apps.facebook.com/keep...)

Programmabl

openkapow

Connect. Collect. Mashup. Everything!

f

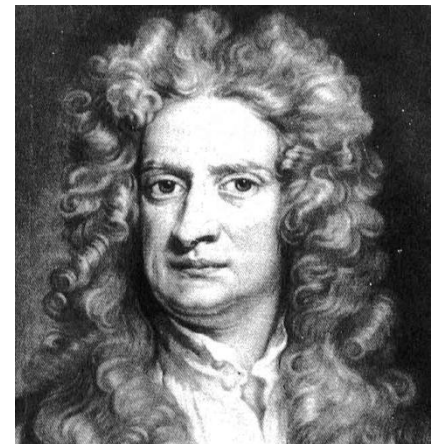
VOICE MASHUPS

THOMAS HOWE

(programmableweb.com)

**"If I have seen further than others,  
it is by standing upon the shoulders of  
giants."**

Isaac Newton, 1642-1727



languages

tools

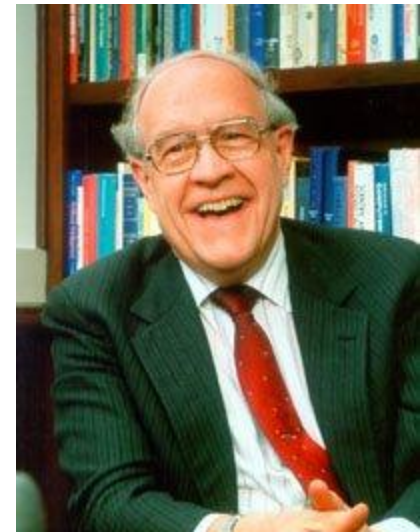
interfaces

components

methodologies

**"There is no single development, in either technology or management technique, which by itself promises even one order-of-magnitude improvement within a decade in productivity, in reliability, in simplicity."**

Fred Brooks, "No Silver Bullet", 1986



# Snake Oil?

- **structured programming**
  - top-down development, successive refinement, chief programmer teams, egoless programming
  - structured everything: design, analysis, requirements, specification, walkthroughs...
- CASE tools (Computer Aided Software Engineering)
  - UML (Unified Modeling Language), message sequence charts, state diagrams
- formal methods
  - verification, validation, correctness proofs, model checking
- **object-oriented programming**
  - CRC cards (Class, Responsibilities, and Collaborators)
  - object-oriented everything: design, analysis, requirements, specification, walkthroughs...
- RAD (rapid application development)
  - **components**, COTS (Components off the Shelf)
  - 4th generation languages, automatic programming, programming by example, graphical programming
- extreme programming, refactoring, agile methods, ...
- aspect oriented programming
- **design patterns**
  - patterns of everything

# Conclusions

- languages
  - notation matters: scripting languages are here to stay
- tools
  - mechanization matters: let the machine do the work
- interfaces
  - interfaces matter: bad APIs are easy; good APIs are hard
- components
  - reuse matters: don't reinvent the wheel
- methodologies
  - knowledge and skill matter: there is no royal road to software

# The Changing Face of Programming??

Plus ça change, plus c'est la même chose