

Towards More Natural Programming for Mobile and Touch

Brad A. Myers

Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University

<http://www.cs.cmu.edu/~bam>

bam@cs.cmu.edu

 Follow @bradamyers

PROMOTO 2015

3rd International Workshop on Programming for Mobile and Touch

27th October 2015, Pittsburgh, Pennsylvania, USA
Co-located with SPLASH 2015

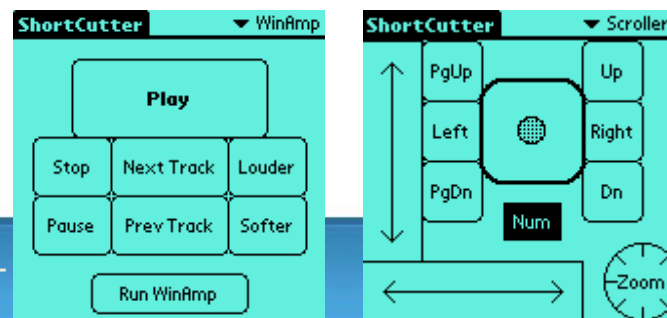


Former Project: Pebbles



Pebbles

- PDAs for Entry of Both Bytes and Locations from External Sources
- <http://www.pebbles.hcii.cmu.edu/>
- One of the first to investigate Personal Digital Assistants (PDAs), 1997-2002
 - Starting with original Palm Pilot, Windows CE 2.1
- Key research – using PDAs *with* PCs
- Provided end-user programming of panels



Natural Programming Project

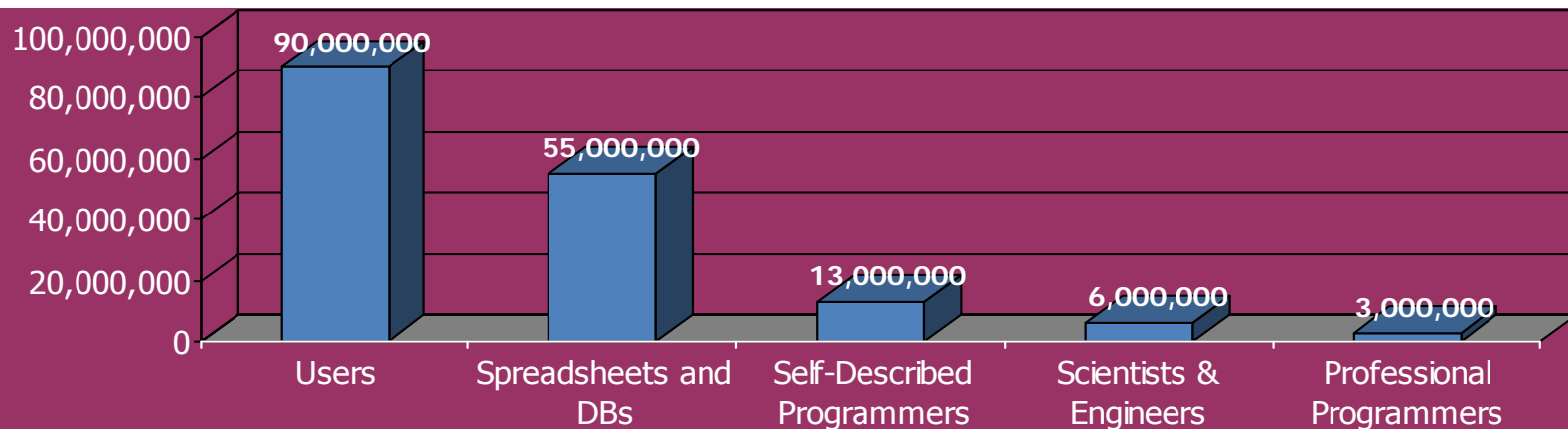


- Researching better tools for programming since 1978
- Natural Programming project started in 1995
- Make programming easier and more correct by making it more *natural*
 - Closer to the way that people think about algorithms and solving their tasks
- Methodology – human-centered approach
 - Perform *studies* to inform design
 - Provide new knowledge about what people do and think, & barriers
 - Guide the designs from the data
 - Design of programming *languages* and *environments*
 - Iteratively evaluate and improve the tools
- Target novice, expert and end-user programmers



“End-User Programmers”

- Programming tools are not just used by highly-trained professional programmers
- **End-User Programmers** = People whose primary job is *not* programming
- In 2012 in USA at work: — [Scaffidi, Shaw and Myers 2005]
 - 3 million professional programmers
 - 6 million scientists & engineers
 - 13 million will describe themselves as programmers
 - 55 million will use spreadsheets or databases at work
 - 90 million computer users at work in US



Why Would Being Natural be Good?



- Programmers are People Too
 - Take the human into account
- Language should be close to user's plan
 - "Programming is the process of transforming a mental plan into one that is compatible with the computer."
— *Jean-Michel Hoc*
- *Closeness of mapping*
 - "The closer the programming world is to the problem world, the easier the problem-solving ought to be.... Conventional textual languages are a long way from that goal." — *Green and Petre*



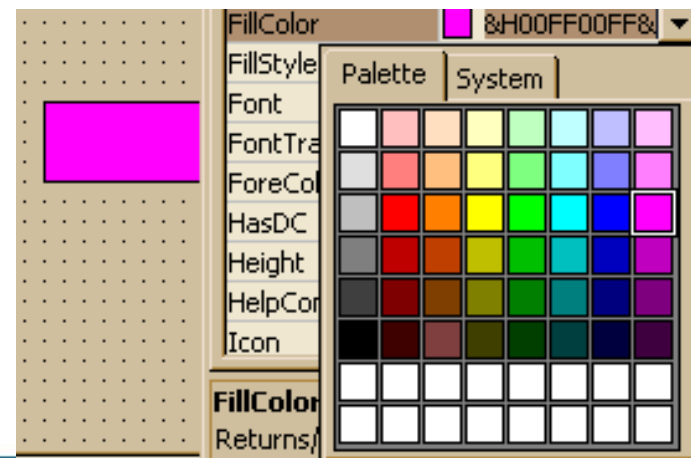
Not so Natural!

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

- 3 kinds of parentheses and 9 special words!
- Compared to click and type: "Hello World!"

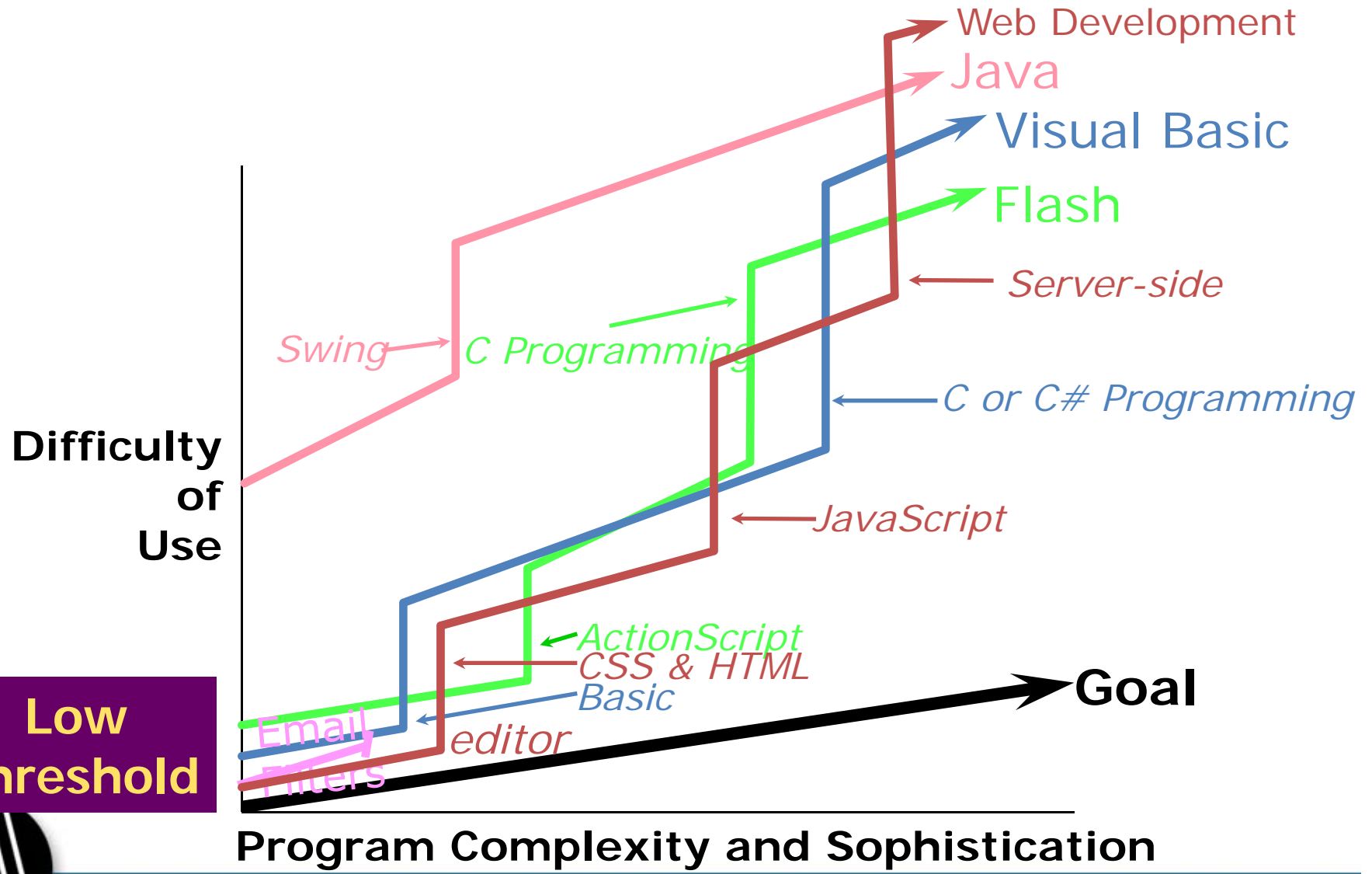


```
Let Shape1.FillColor  
= &H00FF00FF&
```



Goal: Gentle Slope Systems

High Ceiling



UX Techniques to Improve Programming

Field Studies

- Logs & error reports

Exploratory Studies

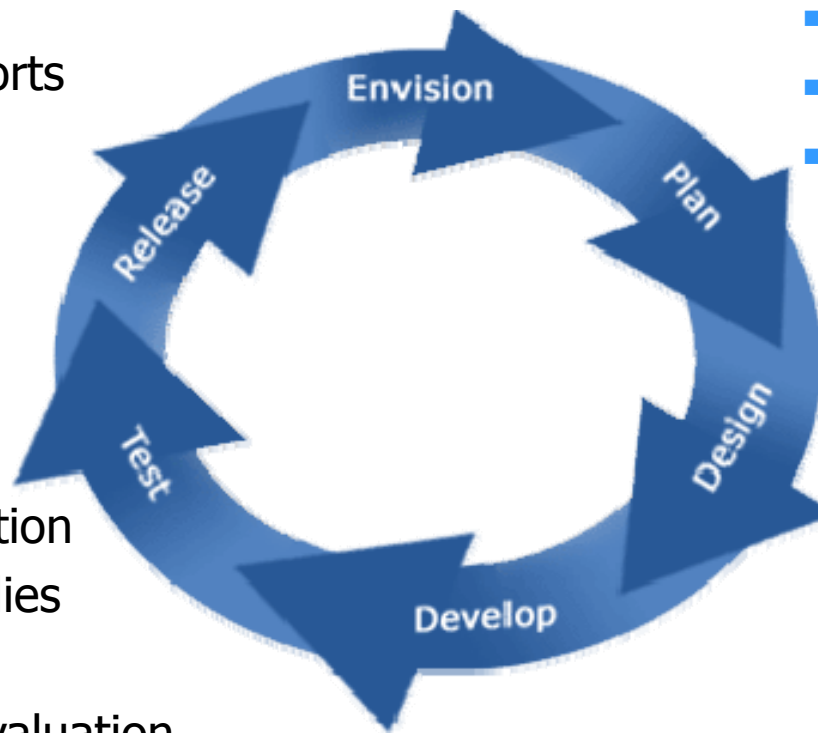
- Contextual Inquiries
- Surveys
- Lab Studies
- Corpus data mining

Evaluative Studies

- Usability Evaluation
- Formal Lab studies
- Expert analyses
 - Heuristic Evaluation
 - Cognitive Walkthroughs
 - Cognitive Dimensions

Design Practices

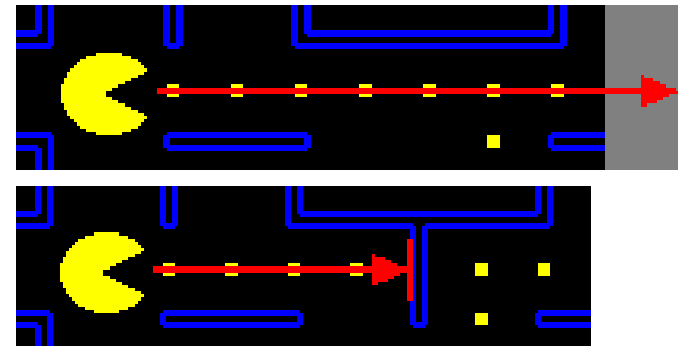
- “Natural programming” elicitation
- Graphic & Interaction Design
- Paper Prototyping



First Natural Programming Studies



- John Pane, PhD 2002
- Studies:
 - How people naturally express programming concepts and algorithms
 - 1) Nine scenes from PacMan
 - 2) Transforming and calculating data in a spreadsheet
 - Specific issue of language design
 - 3) Selecting specific objects from a group (“and”, “or”, “not”)



– Lots of interesting results

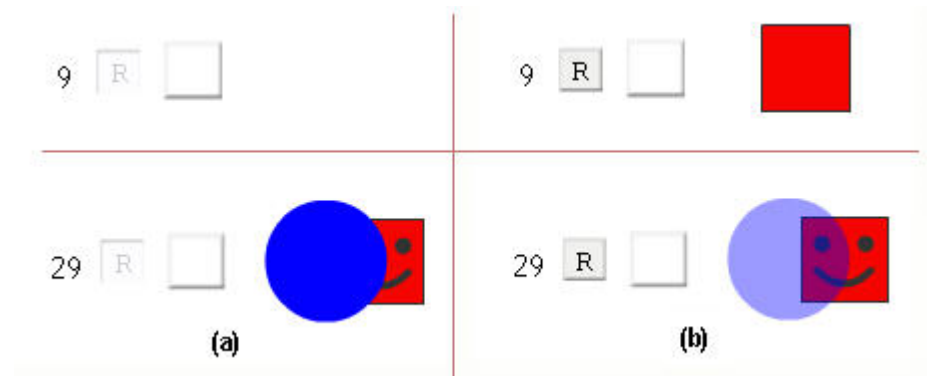
Examples of Results

- Rule-based style
 - “If PacMan loses all his lives, its game over.”
- “And”, “Or”, “Not” don’t match computer interpretation
 - ... left-handed and right-handed people
 - ... (not an apple) or pear
- Operations suggest data as lists, not arrays
 - People don’t make space before inserting
- Objects normally moving
 - “If PacMan hits a wall, he stops.”
 - so objects remember their own state



Interactive Behaviors

- *(VL/HCC'08)*
- Studied natural expression for interactive behaviors & animations
- Before and after pictures of primitives of interactive behaviors
- More use of constraints
- Consistent wording -- "appears", "fades out"



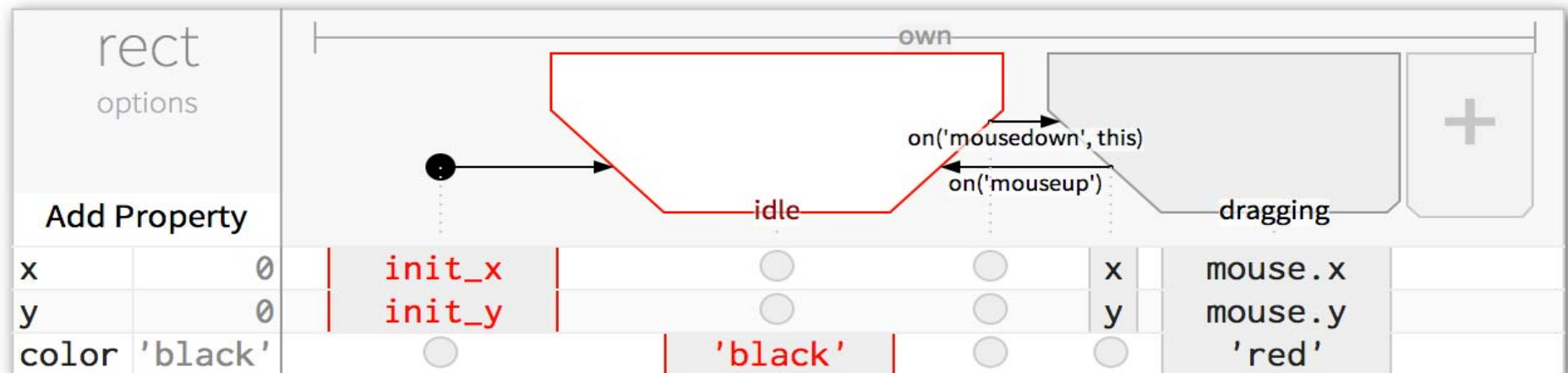


InterState



- PhD work of Stephen Oney (PhD 2015)
 - Now faculty at Univ. Michigan
 - <http://interstate.from.so/>
- Visual Programming Language for expressing behaviors
- Aimed at Interaction Designers (EUPs) who have some experience with programming
- Spreadsheet-like tables for object properties with constraints
- Columns are state machines to control when applied
- Many innovations in language, inheritance model, etc.

(UIST'14)
[Video \(3:36\)](#)



Equivalent drag-lock JavaScript code


```
var isDragLocked = false,
    mm_listener = function(mm_event) {
        draggable.attr({ x: mm_ev.x, y: mm_ev.y });
    },
    mu_listener = function(mu_event) {
        removeEventListener("mousemove", mm_listener);
        removeEventListener("mouseup", mu_listener);
    };
draggable.mousedown(function(md_ev) {
    draggable.attr({ x: md_ev.x, y: md_ev.y });
    addEventListener("mousemove", mm_listener);
    addEventListener("mouseup", mu_listener);
}).dblclick(function(md_event) {
    if(isDragLocked) {
        removeEventListener("mousemove", mm_listener);
    } else {
        addEventListener ("mousemove", mm_listener);
    }
    isDragLocked = !isDragLocked;
});
```

Control
flow on
double-click

Equivalent drag-lock JavaScript code

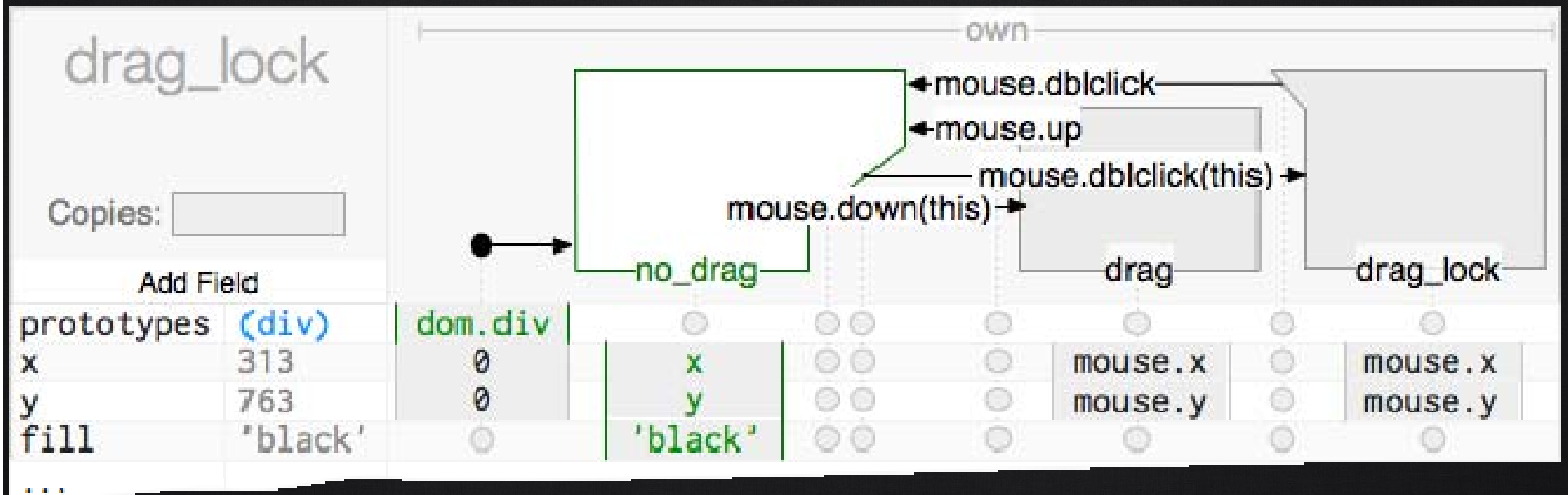
```
var isDragLocked = false,
    mm_listener = function(mm_event) {
        draggable.attr({ x: mm_ev.x, y: mm_ev.y });
    },
    mu_listener = function(mu_event) {
        removeEventListener("mousemove", mm_listener);
        removeEventListener("mouseup", mu_listener);
    };

draggable.mousedown(function(md_ev) {
    draggable.attr({ x: md_ev.x, y: md_ev.y });
    addEventListener("mousemove", mm_listener);
    addEventListener("mouseup", mu_listener);
}).dblclick(function(md_event) {
    if(isDragLocked) {
        removeEventListener("mousemove", mm_listener);
    } else {
        addEventListener ("mousemove", mm_listener);
    }
    isDragLocked = !isDragLocked;
});
```

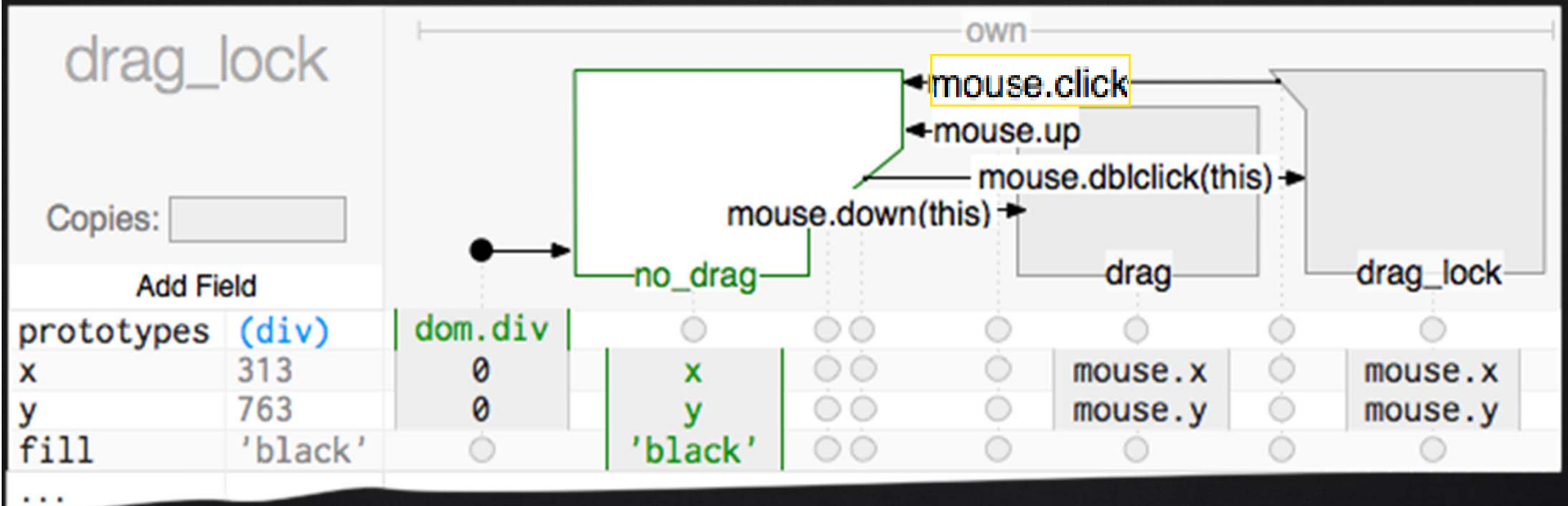


Control
flow when
click again

Most of the InterState code



Changes required for **single**-click to exit



Changes required for **single**-click to exit

```
var isDragLocked = false,
    mm_listener = function(mm_ev) {
        draggable.attr({ x: mm_ev.x, y: mm_ev.y });
    },
    mu_listener = function(mu_event) {
+       if(!isDragLocked) {
            removeEventListener("mousemove", mm_listener);
            removeEventListener("mouseup", mu_listener);
+       }
    };
draggable.mousedown(function(md_ev) {
    draggable.attr({ x: md_ev.x, y: md_ev.y });
+   if(!isDragLocked) {
        addEventListener("mousemove", mm_listener);
        addEventListener("mouseup", mu_listener);
+   }
}).dblclick(function(md_event) {
-   removeEventListener("mousemove", mm_listener);
- } else {
        addEventListener("mousemove", mm_listener);
        isDragLocked = !isDragLocked;
    }
+ }).click(function(c_event) {
+   if(isDragLocked) {
        removeEventListener("mousemove", mm_listener);
        isDragLocked = !isDragLocked;
+   }
+ });
```

InterState: touch clusters

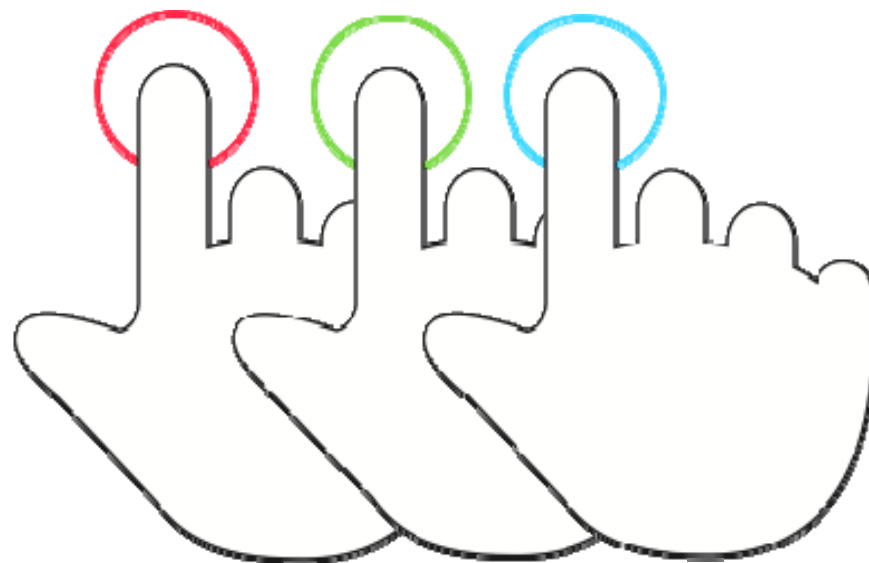
- Newest work – better ways to describe touch events and resulting behaviors
- Developers specify number of fingers, where pressed, etc.
- Outputs: location, scale, rotation
- Resolving conflicts:
 - optional delay & priority for events
 - touch clusters can determine whether other clusters can use same touches



Disambiguation



three-finger
cluster fires




three single-touch
clusters fire

delay	false	false
priority	false	false
eventGroup	false	false
preventDefault	false	false
greedy	false	false
cross	>	



Crossing events



delay	false	false
priority	false	false
eventGroup	false	false
preventDefault	false	false
greedy	false	false
cross	>	

crossEvent

Copies: (empty)

Add Field

prototypes	(cross	touchEvent.
touchCluster	(touch	touchEvent
path	(cross	crossingLin

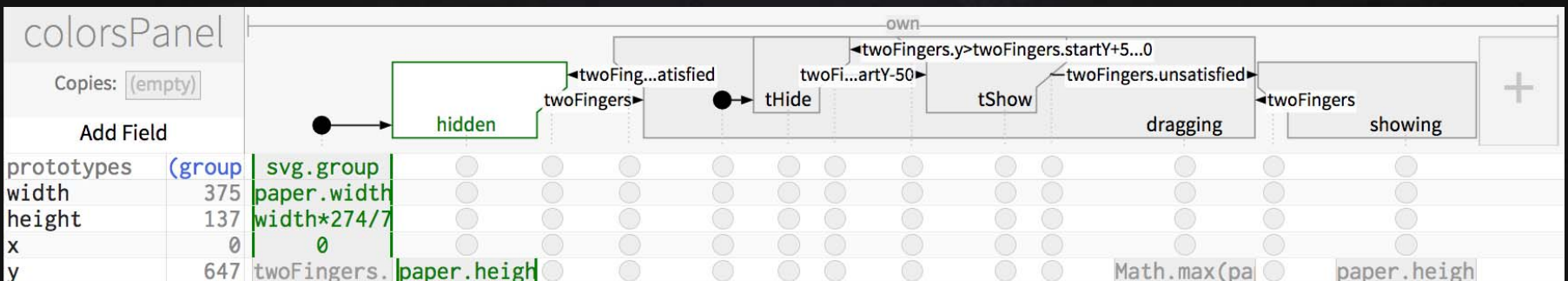
- Lines, circles, rects
- Can be calculated with formulas



Putting these together



- One-finger swipe up for tools
- Two-finger swipe up for colors
- Crossing invisible rectangle at the bottom





Gneiss: Extending Spreadsheet Programming



- PhD work of HCII student Kerry Chang (in progress)
- **Gneiss: Gathering Novel End-user Internet Services using Spreadsheets**
- Extend spreadsheet model so spreadsheet can calculate using web service data, streaming data, and web user interfaces
 - Lists of restaurants, movies, cars, stock prices, RSS feeds, Twitter feeds, ... (almost anything!)
- Can also create user interfaces that use and control the values

*(VL/HCC'14, UIST'14,
CHI'15)*



Gneiss Language



- Code using familiar spreadsheet language
 - Innovation: **pull** (formula) semantics even for user interface elements (instead of events)
- Interface builder to drag in UI elements
 - Connect to spreadsheet cells using formulas
 - Including lists – Autofill-down to populate
- Multiple pages – transitions based on input events and formulas



Gneiss Video



- Right pane could be on mobile device

[Video](#)
(5:00min)

File Edit Setting

on?query={{(A1)}}&key=AlzaSyB6h57

Selected (click edit): results[6] name

When moving the selected item to the spreadsheet, populate the column with similar items in the array.

```

"bar",
"establishment"
}
}
{
  formatted_address : "10 Colum
  geometry :
    {
      location :
        {
          lat : 40.768718,
          lng : -73.982489
        }
      },
      icon : "http://maps.gstatic.c
      id : "94f9a538183c6a67c07e435
      name : "Dizzy's Club Coca Col
      opening_hours :
        {
          open_now : false
        },
      photos :
        [
          {
            height : 960,
            html_attributions :
            photo_reference : "
            width : 1200
          }
        ],
      price_level : 4,
      rating : 4.4,
      reference : "CoQ8eQAAAnsg3EEK
      types :
        [
          "night club",
        ]
      }
    }
  }
}

```

2

A	B (name)	C (rating)	D (price_level)	E (formatted)	F	G
1	Jazz bar New York City Dizzy's Club Coca Cola	4.4	4	10 Columbus Cir #5, New York, NY, United States	false	
2	Little Branch	4.3	3	20 7th Ave S, New York, NY, United States	false	
3	Louis 649	4.2	2	649 E 9th St, New York, NY, United States	true	649 E 9th St, New York, NY, United States
4	Garage Restaurant & Cafe	4.1	2	99 7th Ave S, New York, NY, United States	false	
5	Jazz Standard	4.1	3	116 E 27th St, New York, NY, United States	true	116 E 27th St, New York, NY, United States
6	BIRDLAND	4	3	315 W 44th St, New York, NY, United States	true	315 W 44th St, New York, NY, United States
7	The Flatiron Room	4	3	37 W 26th St, New York, NY, United States	false	
8	La Lanterna di Vittorio	4	2	129 Macdougall St, New York, NY, United States	false	
9	Smoke Jazz & Supper Club	4	2	2751 Broadway, New York, NY, United States	false	
10	BLACK DUCK	4	3	122 E 28th St, New York, NY, United States	false	
11	Knickerbocker Bar & Grill	3.9	3	33 University Pl, New York, NY, United States	false	

3

Places to Go

Jazz bar New York City

Sort descending by rating | Sort descending by price

4

5

Dizzy's Club Coca Cola

Rating: 4.4, Price level: 4

 Plot on map!

Little Branch

Rating: 4.3, Price level: 3

 Plot on map!

Louis 649

Rating: 4.2, Price level: 2

 Plot on map!

Garage Restaurant & Cafe

Rating: 4.1, Price level: 2

 Plot on map!

Jazz Standard

Rating: 4.1, Price level: 3

 Plot on map!

BIRDLAND

Rating: 4, Price level: 3

 Plot on map!

The Flatiron Room

Rating: 4, Price level: 3

La Lanterna di Vittorio

Rating: 4, Price level: 2

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

Gneiss New Features



- Newest work – handle hierarchical data using spreadsheet UIs – e.g., JSON data
- Submitted for publication
- Drag columns to restructure
- Spreadsheet language can refer to cells at multiple levels

1	A (businesses.name)	B (businesses.categories.category)
1	Coca Cafe	1.1 breakfast_brunch 1.2 newamerican
2	Waffles Incaffeinated	2.1 breakfast_brunch 2.2 newamerican 2.3 tradamerican
3	Point Brugge Café	3.1 belgian
4	The Dor-Stop Restaurant	4.1 breakfast_brunch 4.2 diners
5	Deluca's Diner	5.1 breakfast_brunch

2	A (businesses.name)	B (businesses.categories.category)
1	Coca Cafe	1.1 breakfast_brunch 1.2 newamerican
2	Waffles Incaffeinated	2.1 breakfast_brunch 2.2 newamerican

3	A (businesses.categories.category)	B (businesses.name)
1	belgian	Point Brugge Café
2	belgian	Park Bruges
3	breakfast_brunch	Coca Cafe
4	breakfast_brunch	Waffles Incaffeinated

(1) a screenshot of our tool showing a list of restaurants and their categories retrieved from Yelp's JSON web service. Nested tables are used to represent the hierarchical structure. By dragging column B to the front (2), the user reshapes the data and views the restaurants by categories (3).



Study of APIs

Started as PhD work of Jeff Stylos, PhD, 2009

- Inspired by Steven Clarke, Microsoft Visual Studio group
- **A**pplication **P**rogramming **I**nterface
 - Libraries, frameworks, SDKs, ...
- Which programming patterns are most usable?
- Barriers to use of APIs
- Measures: learnability, errors, preferences
- Expert and novice programmers
- Studied:
 - Default parameters in constructors
 - Factory pattern
 - Object design
 - SAP's Web Services APIs
- See www.apiusability.org

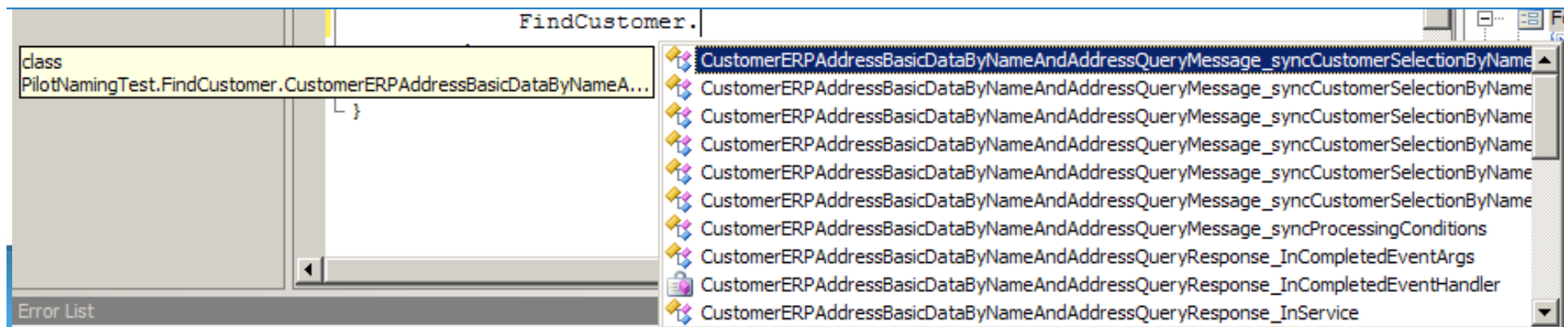


Study of APIs for SAP



- Study APIs for Enterprise Service-Oriented Architectures (“Web Services”)
- Naming problems:
 - Too long
 - Not understandable
 - Differences in *middle* are frequently missed

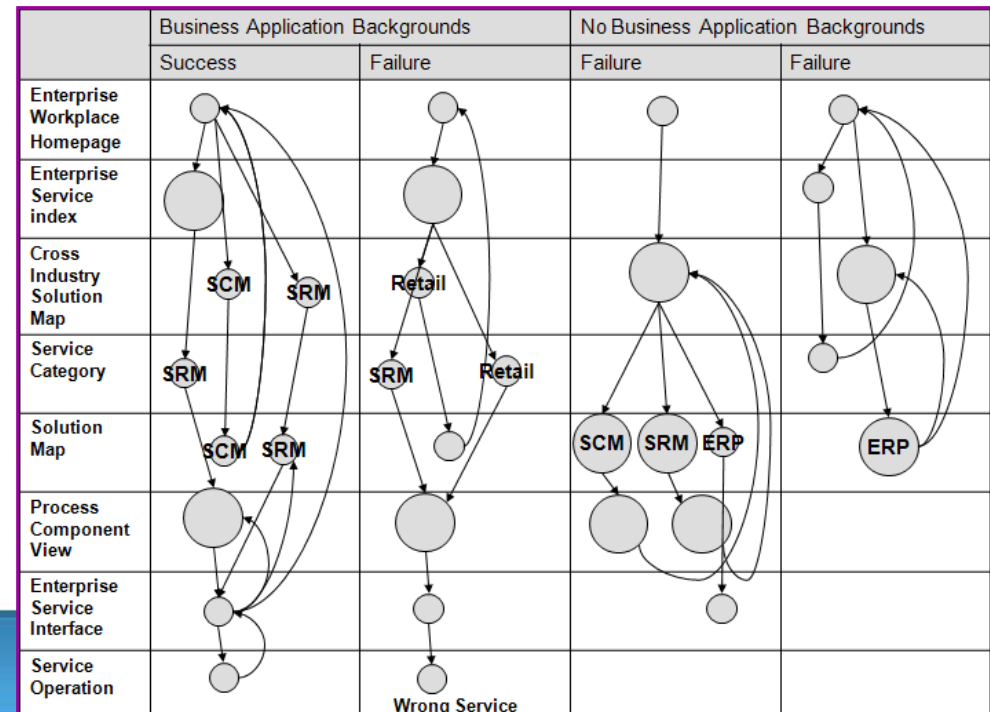
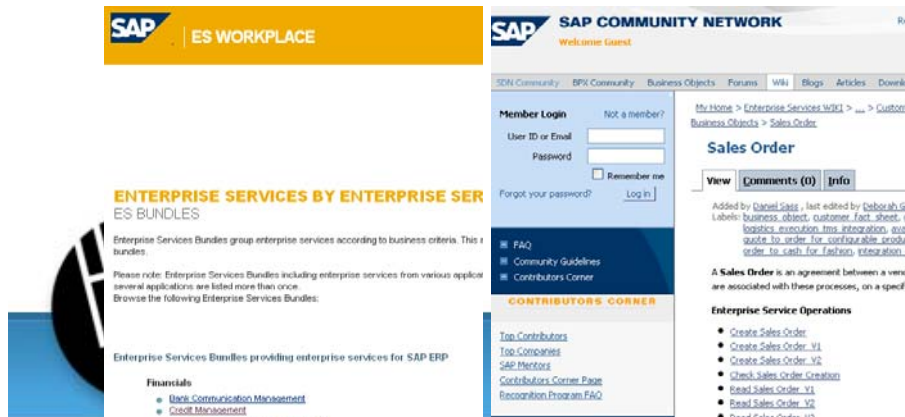
CustomerAddressBasicDataByNameAndAddressRequestMessageCustomerSelectionCommonName
CustomerAddressBasicDataByNameAndAddressResponseMessageCustomerSelectionCommonName



eSOA Documentation Results

- Multiple paths: unclear which one to use
- Some paths were dead ends
- Inconsistent look and feel caused immediate abandonment of paths
- Hard to find required information
- Business background helped

(IS-EUD'2009)



Required Constructors

- Compared **create-set-call** (default constructor)

```
var foo = new FooClass();  
foo.Bar = barValue;  
foo.Use();
```

- vs. required constructors (*immutable* classes):

```
var foo = new FooClass(barValue);  
foo.Use();
```

- All participants assumed there would be a default constructor
- Required constructors interfered with learning
 - Users wanted to experiment with what kind of object to use first
- Preferred to *not* use temporary variables

(Stylos & Clarke, ICSE '2007)



New Project: API Usability & Security

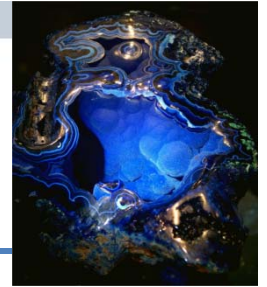


- Collaboration with CMU's Software Engineering Institute (SEI)
- New NSF grant & SEI grant
- Sometimes usability \cong security
 - More usable \rightarrow fewer mistakes
 - E.g., Android and iOS apps misused Secure Sockets Layer (SSL) or Transport Layer Security (TLS) due to difficulties with using the APIs and had vulnerabilities [Fahl, CCS 2013]
- But sometimes usability \neq security
 - Mutability better for usability, worse for security
- How can usability research inform API design for security?
 - Current study: Immutability in APIs – (PhD student Michael Coblenz)
 - Interviews showed `const`, `final`, `readonly`, etc. are inadequate
- Future study: Error and exception handling





Azurite: Exploring Selective Undo



- PhD work of ISR student YoungSeok Yoon (May'2015)
- **Azurite**: Adding **Z**est to **U**ndoing and **R**estoring **I**mproves **T**extual **E**xploration <http://www.cs.cmu.edu/~azurite>
- Work out meaning of selective undo for code
 - Conflicting edits of same region of code
- Time-line visualization of all past operations
- Search through history (time) to find appropriate points *(VL/HCC'13 & '15, ICSE'15)*

The screenshot displays the Azurite IDE interface. The top panel, titled "Code History Diff", shows a comparison between the current version of `MainFrame.java` (lines 28-42) and a previous version from March 12, 2014, at 5:17:13 PM (id:680). The current version contains four lines of code for creating and styling buttons. The previous version contains three lines of code for creating buttons with different labels. The bottom panel, titled "Timeline View", shows a horizontal timeline for `MainFrame.java` and `DrawingEditor.java`. The timeline for `MainFrame.java` is marked with various colored segments (green, blue, red) and vertical lines indicating operations. A vertical line is positioned at 05:17:13 PM, corresponding to the selected version in the diff view. The timeline for `DrawingEditor.java` is currently empty.

Summary of Insights

- Field and lab studies can reveal the real issues
 - Addressing these issues creates tools that are actually useful
- Researcher's intuitions about what might be useful are often wrong
- Our experience highlights:
 - Understanding the barriers can lead to more effective tools
 - Many user-centered methods can be successfully applied to help understand developers and create better tools.
 - Completely different ways to program mobile applications are possible



There are lots of Gemstones!

- *And acronyms are fun!*

Fluorite:

Full of Low-level User Operations Recorded In The Editor



Azurite:

Adding Zest to Undoing and Restoring Improves Textual Exploration



Euklas:

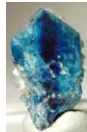
Eclipse Users' Keystrokes Lessened by Attaching from Samples



Apatite:
Associative Perusing of APIs That Identifies Targets Easily

Euclase:

End User Centered Language, APIs System and Environment



Graphite:

GRAPHICAL Palettes Help Instantiate Types in the Editor



Calcite:

Construction And Language Completion Integrated Throughout



Jadeite:

Java API Documentation with Extra Information Tacked-on for Emphasis



Crystal:

Clarifications Regarding Your Software using a Toolkit, Architecture and Language



Jasper:

Java Aid with Sets of Pertinent Elements for Recall



Aquamarine:

Allowing Quick Undoing of Any Marks And Repair Improving Novel Editing



GARNET

Generating an Amalgam of Real-time, Novel Editors and Toolkits



GNEISS:

Gathering Novel End-user Internet Services using Spreadsheets



PEBBLES

PDAs for Entry of Both Bytes and Locations from External Sources



C32

CMU's Clever and Compelling Contribution to Computer Science in CommonLisp which is Customizable and Characterized by a Complete Coverage of Code and Contains a Cornucopia of Creative Constructs, because it Can Create Complex, Correct Constraints that are Constructed Clearly and Concretely, and Communicated using Columns of Cells, that are Constantly Calculated so they Change Continuously, and Cancel Confusion



For more, see: www.cs.cmu.edu/~bam/acronyms.html

Thanks to:



- Funding:

- NSF under CNS-1423054, HCC-1314356, IIS-1116724, IIS-0329090, CCF-0811610, IIS-0757511 (Creative-IT), and NSF ITR CCR-0324770 as part of the EUSES Consortium

- SAP



- Adobe



- IBM



- Microsoft Research RISE



- >32 students:

- | | | |
|----------------------------------|----------------------------|---------------------------------|
| ■ Htet Htet Aung | ■ Erik Harpstead | ■ Yoko Nakano |
| ■ Jack Beaton | ■ Sae Young (Sophie) Jeong | ■ Stephen Oney |
| ■ Ruben Carbonell | ■ Andy Ko | ■ John Pane |
| ■ John R. Chang | ■ Sebon Koo | ■ Sunyoung Park |
| ■ Kerry S. Chang | ■ Ashley Lai | ■ Chotirat (Ann) Ratanamahatana |
| ■ Polo Chau | ■ Thomas LaToza | ■ Christopher Scaffidi |
| ■ Luis J. Cota | ■ Tam Minh Le | ■ Jeff Stylos |
| ■ Michael Coblenz | ■ Joonhwan Lee | ■ David A. Weitzman |
| ■ Dan Eisenberg | ■ Leah Miller | ■ Yingyu (Clare) Xie |
| ■ Brian Ellis | ■ Mathew Mooty | ■ Zizhuang (Zizzy) Yang |
| ■ Andrew Faulring | ■ Gregory Mueller | ■ YoungSeok Yoon |
| ■ Aristiwidya B. (Ika) Hardjanto | | |

Thank You!

Towards More Natural Programming for Mobile and Touch

Brad A. Myers

Human-Computer Interaction Institute

School of Computer Science

Carnegie Mellon University

<http://www.cs.cmu.edu/~bam>

bam@cs.cmu.edu

 Follow @bradamyers

PROMOTO 2015

3rd International Workshop on Programming for Mobile and Touch

27th October 2015, Pittsburgh, Pennsylvania, USA
Co-located with SPLASH 2015