

Professional Java User Interfaces

Mauro Marinilli



John Wiley & Sons, Ltd

Brief Contents

Acknowledgements	xxi
Introduction	xxiii
1 Putting GUI Development into Context	1
Part I User Interface Design	
2 Introduction to User Interface Design	31
3 Java GUI Design	77
4 Recurring User Interface Designs	119
Part II Software Design	
5 Iterative GUI Development with Java	169
6 Implementation Issues	223
7 Code Organization	287
8 Form-Based Rich Clients	323
9 Web-Based User Interfaces	359
10 J2ME User Interfaces	375
11 Java Tools and Technologies	393
12 Advanced Issues	447
Part III Examples	
13 Rich Client Platforms	471
14 The Personal Portfolio Application	497
15 An Example OO User Interface	535

16	An Example Ad-Hoc Component	567
A	A Questionnaire for Evaluating Java User Interfaces	607
B	A Questionnaire for Evaluating J2ME Applications	613
	References	621
	Index	629

Contents

Acknowledgements	xxi
Introduction	xxiii
The interactivity thrill	xxiii
Usable GUIs and usable books	xxiv
The organization of the book	xxv
Three levels of advice	xxvii
Conventions used in the book	xxviii
Source code	xxviii
Reader feedback	xxix
Book readers and personas	xxix
Lars, a Java intermediate programmer	xxx
Keiichi, a tech lead	xxx
Shridhar, a professor in computer science	xxxi
Melinda (Mellie), a manager	xxxi
William, a first year student in a Master in CS course	xxxii
Karole, a business analyst	xxxiii
Juan, an experienced programmer	xxxiii
1 Putting GUI Development into Context	1
1.1 Introduction	2
1.2 Focusing on users	2
1.3 A functional decomposition for user interfaces	3
1.4 Tool selection: the Java singularity	6
Of running little green men and wrong choices	7
1.5 Organizational aspects	7
People and GUIs	8
Team composition	10
1.6 Early design	12
Use case diagrams and GUIs	12

1.7	Lifecycle models, processes and approaches	14
	Rational Unified Process	15
	Extreme Programming and other Agile approaches	20
	LUCID methodology	22
	Evolutionary Prototyping process	24
1.8	UML notation	26
	Class diagrams	26
	Sequence diagrams	28
	State diagrams	29
1.9	Summary	30

Part IV User Interface Design

2	Introduction to User Interface Design	31
2.1	The human factor	32
	A model of interactive systems – seven stages and two gulfs	32
	Developers are part of the design process	34
	Short term memory and cognitive modeling	36
	Interacting with human beings	39
	User-centered design	44
	Simplified thinking aloud	46
2.2	Display organization	47
	Esthetic considerations	49
	Abstract-Augmented Area for GUIs	51
2.3	Interaction styles	57
	Menu selection	57
	Form filling	59
	Language-based styles	62
	Direct manipulation	63
2.4	Conceptual frameworks for UI design	64
	Entity-based approaches to UI design	65
	Metaphor-based approaches to UI design	66
	Function-based approaches to UI design	67
	‘Null’ approach to UI design	67
	Object-oriented user interfaces	69
2.5	Assessing the quality of a GUI	72
	Usability heuristics	73
2.6	Summary	75

3	Java GUI Design	77
3.1	Java technology for GUIs	77
	Assembling the components	77
	Three levels of component cost	79
3.2	Cost-driven design	81
	Ad-hoc versus custom – the difference between ‘run’ and ‘ride’	81
	When ad-hoc is the only way to go	83
3.3	Exploring the design space for a point chooser	86
	Standard designs	88
	Ad-hoc designs	91
	Mixed designs	97
	Conclusions	99
3.4	Design guidelines for the Java platform	100
	Introduction to the guidelines	100
	J2SE user interface design guidelines	103
3.5	The Java look and feel design guidelines	108
	Some definitions	109
	The Java ‘look’	109
	The Java ‘feel’	110
	Some terminology	112
	An example – applying the guidelines for designing dialogs	114
3.6	Summary	117
4	Recurring User Interface Designs	119
4.1	GUI area organization	120
	Terminology	120
	Main frames	121
	Multiple document interfaces	123
	Wizards	124
4.2	Choosers	126
	Chooser activation mechanisms	127
	Chooser interaction styles	128
	Broadening the choice	131
	Conclusions	133
4.3	Memory components	133
	Input history	134
	Saving user preferences	134
4.4	Lazy initialization	135

4.5	Preference dialogs	137
	Preference dialogs styles	138
4.6	Waiting strategies	141
4.7	Flexible layout	144
4.8	Common dialogs	146
	The 'About' dialog	146
	Log-in dialog	149
	First-time message dialogs	150
	Splash window	151
4.9	Command components	152
	Graphic conventions	156
	Toolbar composition	157
	Command composition	158
4.10	Accessibility	160
	Testing the final product for accessibility	161
	Conclusions	161
4.11	Navigation and keyboard support	161
	Keyboard shortcuts	161
	Tab traversal	162
4.12	Internationalization	163
4.13	Help support	164
4.14	Icons and images	165
4.15	Leveraging object-oriented programming	166
4.16	Summary	167

Part V Software Design

5	Iterative GUI Development with Java	169
5.1	Iterating wisely	171
5.2	Introduction to prototyping	173
	Uses for prototyping	174
	The two dimensions of prototyping	176
	Competitors' product as ready-made prototypes	177
	Prototyping as a philosophy for development	177
	Prototypes and customers	178
5.3	Prototyping alternatives	179
	Different types of prototypes	179
	Storyboards	182

5.4	GUI builders	184
5.5	Reusable prototyping widgets	187
	A tree prototype utility class	187
	A visual container prototype utility class	190
5.6	GUI refactoring	191
	Some classic refactorings	191
	Some GUI-specific refactorings	194
	Failing with style	201
5.7	Introduction to user interface testing	203
	Test-driven development	204
	What's first – GUI design or implementation?	204
5.8	Software testing of Java GUIs	206
	How to test – GUI software test approaches	206
	What to test – test coverage criteria	210
5.9	Usability testing of Java GUIs	211
5.10	JRE runtime management	214
	Introduction to profiling	214
	Common problems	216
	Continuous profiling	219
	A posteriori profiling	220
5.11	Summary	221
6 Implementation Issues		223
6.1	Revisiting the abstract model	224
	Testing the various layers	226
	The principle of Single Functional Responsibility	227
	Isolating presentation details	228
6.2	Content	229
	Content assembly	229
	Explicit navigation	231
6.3	Business domain	231
6.4	Data input-output	233
	A comprehensive data IO design strategy	233
	Some design patterns	234
	Remote communication design	235
	Security issues	236
6.5	Making objects communicate	239
	The Observer pattern	240
	Swing events	242

	SWT events	243
	Design-time class decoupling with events	243
	Event Arbitrator	245
	Misuses of event-based messaging	248
	Alternatives to event-based communication mechanisms	250
6.6	Separating data from views	251
	Model-View-Controller	252
6.7	Interaction and control	257
	Representing user actions with the Command pattern	257
	Control issues	260
	A state-oriented approach to GUI control	265
6.8	Some design patterns for GUIs	271
	Adaptation	271
	Composite Context	277
	Active Object	279
	Object lifecycle management – a general mindset	280
	Value Model	282
6.9	GUI complexity boosters	283
6.10	Summary	285
7	Code Organization	287
7.1	Introducing software architectures	287
	Taming references	289
	Composable units	292
	Evolving order and appropriate architectures	294
7.2	Some common GUI architectures	295
	The smart GUI antipattern	296
	A semi-smart GUI architecture	297
	A three-layer architecture	298
	A four-layer architecture	299
7.3	A three-layer organization for GUI code	300
	Overview	301
	The presentation layer	303
	The application layer	304
	The service layer	305
7.4	Two examples of a three-layer implementation	306
	An MP3 player	306
	An electronic circuit simulator and editor	309

7.5	The service layer	314
	Overview	314
	Persistence services	319
	Factory services	319
	Other services	320
7.6	Summary	321
	Key ideas	322
8	Form-Based Rich Clients	323
8.1	Introduction	323
	Defining rich clients	324
	Java rich clients	325
	GUI design for rich clients: the Third Way	326
8.2	Reference functional model	326
	Distributing behavior between client and server	327
	Common problems	329
8.3	Runtime data model	329
	Validation	332
	When to validate and notify	339
8.4	The cake-ordering application, the XP way	343
	Setting up the first Iteration	344
	Content first	346
	Data second	350
	Commands third	353
	Closing the loop with the server	357
8.5	Summary	358
9	Web-Based User Interfaces	359
9.1	An overview of Web user interfaces	359
9.2	GUI design for the Web	361
	Fine graphics details	361
	Area organization	362
	Levels of client-side control	362
	Navigation issues	365
9.3	Implementing Web applications with Java	368
	The typical architecture of a Web application	368
	Basic Java Web GUI technologies	369
	Java applets	370

9.4	From Web applications to rich clients	371
	Different development habits	372
9.5	Summary	373
10	J2ME User Interfaces	375
10.1	Introduction to the MID profile	375
	Main UI concepts	376
	Main UI limitations	379
	Cost-driven design for J2ME GUIs	379
10.2	The MIDP UI API	379
	UI widgets	379
10.3	Designing MIDP GUIs	382
	Abstract GUI designs	384
10.4	Designing navigation	385
10.5	An example custom item	387
10.6	An example ad-hoc item	388
10.7	An example application	389
	The code	391
10.8	Summary	392
11	Java Tools and Technologies	393
11.1	Introduction to tool selection	394
11.2	Evaluating open source software	394
	Open Source Maturity Model	396
11.3	SWT or Swing?	397
	The toolkits	397
	Choosing a toolkit	403
	Mix and match	409
11.4	Other GUI technologies	410
11.5	Utility libraries	411
	Security tools	412
	Deployment tools	413
	Glazed Lists	413
	JGoodies Swing Suite	414
	L2FProd Common Components	415
	Other OSS component libraries	415
	Some commercially-available Swing components	418
11.6	Test tools	420
11.7	Profiling tools	421

11.8	GUI builders	422
11.9	Presentation layer technologies	424
	Assessing a look and feel	425
	Swing look and feels	426
	SWT Presentation	442
11.10	Declarative GUIs with Java	443
	XML-based formats	443
11.11	Summary	445

12 Advanced Issues 447

12.1	Building on top of existing libraries	447
	Attributes	448
	Roll your own framework	451
12.2	Memory management for complex GUIs	454
	A practical case	455
12.3	Restructuring existing GUI code	457
	Porting an old applet – a case study	459
	Long-life GUIs	463
	Providing new deployment support	464
12.4	Exploiting technology	465
12.5	Domain-specific and Little languages	466
12.6	The future of Java GUIs	468
12.7	Summary	469

Part VI Examples

13 Rich Client Platforms 471

13.1	Introduction to Java rich client platforms	471
	The case for RCP applications	472
	What’s in an RCP	473
	GUI design guidelines and RCPs	474
13.2	The NetBeans RCP	474
	NRCP architecture	475
13.3	The Spring RCP	476
13.4	The Eclipse RCP	477
	Eclipse plug-in architecture	477
	Eclipse RCP plug-ins	478
	The workbench – the building blocks of ERCP GUIs	479
	GUI design guidelines for ERCP applications	482

13.5	Choosing the best RCP for your needs	483
	When to employ an RCP	484
13.6	Legal issues	485
	Eclipse	485
	Netbeans	486
13.7	An example Eclipse RCP application	486
	The application	487
	Introducing client-side modular architectures	489
	The Snooper application architecture	492
13.8	Summary	496
14	The Personal Portfolio Application	497
14.1	The scenario	497
	A note on lifecycle models	498
14.2	Analysis	499
	Early analysis	499
	Some scenarios	503
	A refined use case diagram	504
	Individuating boundary classes	505
14.3	Choosing a technology	505
14.4	An initial GUI design	506
	An initial GUI paper mock-up	506
	A second GUI paper mock-up	508
	A throw-away GUI prototype	511
	Validating the throw-away prototype	512
14.5	The final GUI	513
14.6	Implementation	515
	Software requirements	516
	The software architecture	516
14.7	Resources	524
	Localization bundles	525
	Images	525
14.8	The code	525
	The remote explorer director	526
	The prototype	529
14.9	Deployment issues	529
	Server support	529

14.10	An alternative, cost-driven implementation	530
	Choosing a higher-level starting point	530
	A cost-driven prototype using JDNC	531
	A brief introduction to JDNC	531
14.11	Summary	533
15	An Example OO User Interface	535
15.1	Introduction	536
	A matter of style	536
15.2	Implementing object-oriented user interfaces	537
	The Viewable interface	538
15.3	Some utility classes	542
	Brief views	542
	Making collections viewable	543
15.4	Configuration views	544
	A utility class	544
15.5	Interacting with the user	546
	The Commandable interface	546
15.6	Managing user commands	549
15.7	An example application	550
	OOUI objects	551
	The code	555
	Libraries	558
	Some GUI design considerations	559
	Control issues	560
15.8	An alternative implementation using Naked Objects	563
15.9	Summary	565
16	An Example Ad-Hoc Component	567
16.1	Introduction	568
16.2	The Drawing Sandbox application	569
	The application	570
16.3	The Sandbox architecture	573
16.4	The Sandbox component	575
	Top-down refinement of functional organization	575
	Organizing object communication	575
	Graphical objects	579

16.5	User interaction	582
	Command composition	582
	The Action framework	583
	The Actions class	584
	Undo-redo support	587
	The Edit class	588
	Recording edits	589
	Memory issues	590
16.6	Control	592
	The Director class	594
	Managing actions	595
	Enforcing logical constraints on actions	596
16.7	The whole picture	597
16.8	Stressing the software design	598
	Adding objects and commands	599
	The design's weak points	600
16.9	Introducing JHotdraw	601
16.10	Summary	605
	Key ideas	605
A A Questionnaire for Evaluating Java User Interfaces		607
B A Questionnaire for Evaluating J2ME Applications		613
References		621
	General advice on usability and GUI design	626
	Java-specific links	627
Index		629

Acknowledgements

I have been working on this book for more than five years, in one way or another. It is by no means the result of a single person (the author). A very large number of people shaped it, so many that it will be impossible to name them all.

I should first thank Sally Tickner and the management at John Wiley and Sons, for allowing me to deliver the manuscript after such a huge delay, which was mostly caused – only a partial excuse – by my never-ceasing joy of working on and taming new and complex adventures, rather than lack of interest in the subject.

It's easy to remember Steve Rickaby of WordMongers, whose expert hand made this book readable and sometimes even enjoyable. If it wasn't for his help this gigantic work would have been very different.

A special thanks too to my students, who kept the fire of honest enthusiasm and the Spirit of Wonder high, and who were extremely patient with my shameful schedule. I hope I gave them back at least a small portion of what they gave to me.

I wish also to thank the reviewers for their careful work, the many clients and colleagues for the countless lessons that I tried to put together in the book, and all the people who spent precious time out of their lives putting into written form their hard-won experience, often without any economic return.

Last but not least, my biggest thanks go to my family, my Bella, and my closest friends. Without their presence, patience, and constant support, this book wouldn't have been possible.