

K. Gottheil H.-J. Kaufmann
Th. Kern R. Zhao

X und Motif

Einführung in die Programmierung des
Motif-Toolkits und des X-Window-Systems

Mit 138 Abbildungen und 12 Tabellen



Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Inhaltsverzeichnis

1	Einleitung	13
1.1	Das Motif-Toolkit	13
1.2	Lesehinweise	15
1.3	Weiterführende Literatur	16
A	Überblick und Konzepte	19
2	Fenster zur Welt	21
2.1	Wozu Windows?	21
2.2	Window-Historie	24
3	X-Windows machen's möglich	29
3.1	Server und Client	29
3.2	Windows	31
3.3	Grafik im Raster	33
3.3.1	Grundlagen der Rastergrafik	33
3.3.2	Grafikausgaben mit der Xlib	36
3.4	Ereignisse werfen ihre Schatten voraus	38
3.5	Wunschlisten für Benutzer	40
3.6	Ein kleines Beispiel	41
4	Das Motif-Toolkit	49
4.1	Die Begriffswelt der Widgets	50

4 Inhaltsverzeichnis

4.1.1	Klassen und Instanzen	51
4.1.2	Ein Baum von Klassen	52
4.1.3	Widgets, Windows und Shells	53
4.2	Widgets konkret	57
4.2.1	Das Hauptprogramm von <i>xtime</i>	58
4.2.2	Die Widgets erzeugen	61
4.2.3	Den Dialog organisieren	66
4.2.4	Der letzte Schliff	69
B	Programmieren mit Motif	71
5	Arbeiten mit X	73
5.1	Starten von X	74
5.1.1	Server, Clients und Displays	74
5.1.2	Die Maus	75
5.1.3	Das Programm <i>xinit</i>	76
5.1.4	Erweiterungen für <i>.login</i> und <i>.cshrc</i>	78
5.2	Terminals im Fenster	79
5.3	Schneiden und Kleben	81
5.4	Fenstergeometrie	82
5.5	Farben	83
5.6	Zeichensätze	85
5.7	Ressourcen	87
5.8	Ressourcen in den Server laden	94
5.9	Motif-Window-Manager	95
5.9.1	Starten des <i>mwm</i>	96
5.9.2	Mwm-Ressourcen	98
5.9.3	Menüs	99
5.9.4	Funktionen	99
5.9.5	Beispiel für ein Konfigurations-File	99
5.9.6	Im Brennpunkt: der Tastaturfokus	107
5.9.7	Icons selbstgemacht	107
6	Programmieren mit Xt und Motif	109
6.1	Hello World zum zweiten...	109
6.1.1	Namenskonventionen und Bibliotheken	111
6.1.2	Include-Files	113
6.1.3	Initialisierung	114
6.1.4	Einfache Fehlerbehandlung	116
6.1.5	Keine Applikation ohne Shell	117
6.1.6	Das <i>Label</i> -Widget	118
6.1.7	Realisierung einer Widget-Hierarchie	123

6.1.8	Ereignisse ohne Ende: die Event-Schleife	123
6.2	Initialisierung mit Komfort	124
6.3	Der Default-Application-Context	126
6.4	Optionen	127
6.5	Wichtige Datentypen	131
6.6	Argumentlisten	132
6.7	Mehr über Argumentlisten	135
6.8	Erzeugen und Zerstören von Widgets	139
6.9	Callbacks	142
6.10	Variable Parameterlisten	149
7	Der Aufbau von Formularen	151
7.1	Das Beispiel „Software-Entwicklung“	151
7.1.1	Bedienung und Aufbau	152
7.1.2	Dialogobjekte und Layout-Widgets	153
7.1.3	Ein ganz normales Hauptprogramm	154
7.1.4	Ein einzeliges Formular	156
7.1.5	Geschachtelte Formulare	159
7.1.6	Callback-Routinen für die Kommandos	162
7.2	Allgemeines zur Beschreibung von Widget-Klassen	170
7.3	Geometrie und Farben der Dialogobjekte	172
7.4	Das <i>Label</i> -Widget	176
7.4.1	Compound-Strings im <i>Label</i> -Widget	176
7.4.2	Bilder im <i>Label</i> -Widget	181
7.4.3	Ausrichten des Label-Inhaltes	183
7.4.4	Trennen mit einem <i>Separator</i> -Widget	185
7.5	<i>PushButton</i> -Widgets	186
7.5.1	Verschiedene Callback-Listen	186
7.5.2	Das Erscheinungsbild eines Push-Buttons	190
7.5.3	Ein Widget abschalten: Die Ressource <i>XmNsensitive</i>	192
7.5.4	Vorgefertigte Buttons	193
7.6	Der <i>Toggle</i> -Button	195
7.6.1	Zustand eines <i>Toggle</i> -Buttons	195
7.6.2	Zustandsänderungen im Programm	197
7.6.3	Markierungen und <i>Radio</i> -Buttons	198
7.7	Das <i>Text</i> -Widget	201
7.7.1	Bedienung eines <i>Text</i> -Widgets	201
7.7.2	Der Text im Widget	202
7.7.3	Das Aussehen des <i>Text</i> -Widgets	204
7.7.4	Die Callback-Routinen eines <i>Text</i> -Widgets	205
7.7.5	Ein einfacher Text-Editor	207
7.7.6	<i>Text</i> -Widgets mit Scrollbars	211
7.7.7	<i>Text</i> -Widgets für Fortgeschrittene	212
7.7.8	Eine spezielle Widget-Klasse für einzeilige Texte	216

7.8	Plazieren mit dem <i>RowColumn</i> -Widget	216
7.8.1	Die verschiedenen Varianten eines <i>RowColumn</i> -Widgets	217
7.8.2	Freies Plazieren im <i>RowColumn</i> -Widget	218
7.8.3	Kinder einheitlicher Größe	219
7.8.4	Kinder gleicher Höhe oder Breite	220
7.8.5	Einheitliche Ressourcen für die Kinder	222
7.9	Ein universelles Layout-Widget, das <i>Form</i> -Widget	223
7.9.1	Constraint-Ressourcen	223
7.9.2	Plazierung der oberen Kante eines Kindes	227
7.9.3	Kinder flexibler Größe	228
7.10	<i>Form</i> -Widgets im Einsatz	230
7.10.1	Ein <i>Form</i> -Widget anstelle eines <i>RowColumn</i> -Widgets	230
7.10.2	Dehbare Eingabezeilen	235
7.10.3	Layout im Ressourcen-File	240
7.10.4	Tabellen mit dem <i>Form</i> -Widget	243
7.11	Die Superklassen der Layout-Widgets	243
7.11.1	Layout-Widgets im Motif-Toolkit	243
7.11.2	Verschiedene Maßeinheiten	245
7.11.3	Schatten am Layout-Widget	246
8	Formulare im Zusammenspiel	247
8.1	Shell-Widgets als „Tor“ zum Window-Manager	249
8.1.1	Ende nur auf Bestätigung	249
8.1.2	Dialoghierarchie	251
8.1.3	Shell-Hierarchie und Popup-Shells	253
8.1.4	Klassenhierarchie der Shell-Widgets	255
8.1.5	Zusammenspiel mit dem Window-Manager	258
8.1.6	Spezielle Ressourcen für den Fensterrahmen	259
8.1.7	Tastaturfokus und Tab-Groups	261
8.1.8	Begriffe	265
8.2	Meldungen für den Benutzer	266
8.2.1	Die Bestandteile einer Message-Box	266
8.2.2	Verschiedene Formen einer Message-Box	268
8.3	Das Bulletin-Board	270
8.3.1	Das Bulletin-Board als Metaklasse für Layout-Widgets	271
8.3.2	Bulletin-Boards und Shell-Widgets	271
8.3.3	Modale Formulare	273
8.3.4	Rahmen, Buttons und Fonts	273
8.4	Mehrere Formulare mit gleichem Aufbau	274
8.4.1	Interaktive Erzeugung eines Formulars	276
8.4.2	Das Formular und die Formulardatenstruktur	279
8.4.3	Eine Shell wird zerstört	280
8.4.4	Hierarchien	283
8.4.5	Mehrere Application-Shells	285

8.5	Dateien auswählen	286
8.5.1	Zur Bedienung einer File-Selection-Box	286
8.5.2	Die Widget-Klasse <i>XmFileSelectionBox</i>	288
8.5.3	Eine File-Selection-Box für jedes Arbeitsformular	291
8.5.4	Toggle-Buttons zur Steuerung des Datenflusses	296
8.6	Strings auswählen	299
8.6.1	Die Widget-Klasse <i>XmSelectionBox</i>	299
8.6.2	String-Listen und Callbacks der Selection-Box	301
8.6.3	Anwendungen einer Selection-Box	303
8.6.4	Die Work-Area einer Selection-Box	305
8.6.5	Prompt-Dialog-Box als leere Dialogbox	306
8.6.6	Die Widget-Klasse <i>XmList</i>	307
8.6.7	Verschiedene Selektionsarten	312
8.6.8	Ein <i>List</i> -Widget mit Scrollbars	315
8.7	Das <i>Scrollbar</i> -Widget	317
8.7.1	Widgets mit dem Scrollbar verschieben	317
8.7.2	Eine universelle Zeichenfläche	319
8.7.3	Die Länge des Balkens	320
8.7.4	Ein Widget verschieben	322
8.7.5	Andere Widgets mit Scrollbars	323
8.7.6	Weitere Ressourcen eines Scrollbars	324
8.7.7	Zahlen auf einer Skala darstellen	325
8.8	Aspekte der Fehlerbehandlung	326
8.8.1	Vorsorge gegen Fehlbedienung	326
8.8.2	Fehler im <i>Text</i> -Widget	328
8.8.3	Jede Änderung im Text prüfen	331
8.8.4	Fehlermeldungen aus dem Ressourcen-File	332
8.8.5	X-Toolkit- und Xlib-Fehler	339
8.8.6	Die lokale Event-Schleife – einfach, aber unsauber	344
8.9	Die letzte Version des Programms <i>soft</i>	351
8.10	Auf der Suche nach Widgets und Ressourcen	353
8.10.1	Die richtige Widget-Klasse	353
8.10.2	Die Ressourcen für den gewünschten Effekt	355
8.10.3	Widgets ohne Window: Gadgets	355
9	Grafik	359
9.1	Der Grafik-Testrahmen <i>grtest</i>	361
9.2	Farben	368
9.2.1	Was kann der Bildschirm?	371
9.2.2	Farben erfragen	375
9.2.3	Öffentliche Einträge in der Farbtabelle	376
9.2.4	Private Einträge in der Farbtabelle	378
9.2.5	Einträge in der Farbtabelle freigeben	381
9.2.6	Farbtabellen einrichten	382

9.3	Muster	383
9.3.1	Pixmap erzeugen und zerstören	384
9.3.2	Pixel-Daten in der Applikation bearbeiten	385
9.3.3	Bitmap-Files	386
9.4	Buchstaben	389
9.4.1	Fonts suchen	389
9.4.2	Fonts laden und wieder freigeben	390
9.4.3	Informationen über Fonts einholen	391
9.5	Grafikkontexte	391
9.5.1	Grafikkontexte verwalten	394
9.5.2	Attribute von allgemeinem Interesse	396
9.6	Die Grafikressourcen für <i>grtest</i>	403
9.7	Grafik ausgeben	412
9.7.1	Linien zeichnen	413
9.7.2	Flächen füllen	419
9.7.3	Text schreiben	423
9.7.4	Pixmap ausgeben	428
9.7.5	Flächen löschen	430
9.8	Die Grafikprimitiven für <i>grtest</i>	430
9.9	Der Cursor	434
10	Grafikdaten verwalten	437
10.1	Grafikdaten in Listenform	439
10.2	Grafikdaten als Objekte	445
10.2.1	Objektorientierte Programmierung	445
10.2.2	Objekte und Klassen in C	447
10.2.3	Die grafische Datenstruktur	451
11	Ereignisse intim	475
11.1	Datentypen für Events	476
11.2	Events auswählen	478
11.3	Event-Typen im Detail	479
11.3.1	Tastatur- und Maus-Events	480
11.3.2	Window-Eintritt und -Austritt	484
11.3.3	Expose-Events	486
11.3.4	Tastaturfokus	488
11.3.5	Window-Änderungen	488
11.3.6	Sonstige Events	488
11.4	Der Grafikeditor <i>gredi</i>	493
11.5	Events selbst behandeln	505
11.6	<i>gredi</i> mit Event-Handlern	507
11.7	Actions	512
11.7.1	Action-Routinen bekannt machen	513

11.7.2	Translation-Tables in der Applikation ändern	515
11.7.3	Eigene Translation-Tables schreiben	515
11.7.4	Tasten umcodieren	524
11.8	<i>gredi</i> mit Action-Routinen	525
11.9	Eingaben beschleunigen	530
12	Menüs à la carte	533
12.1	Pulldown-Menüs	533
12.1.1	Pulldown-Menüs im Detail	534
12.1.2	Ein einfaches Pulldown-Menü	538
12.1.3	Hierarchische Pulldown-Menüs	544
12.2	Popup-Menüs	548
12.2.1	Erzeugung einer Popup-Pane	549
12.2.2	Testrahmen für Popup-Menüs	550
12.3	Option-Menüs	553
12.3.1	Der Aufbau eines Option-Menüs	553
12.3.2	Ein Beispiel für das Option-Menü	555
12.4	Beschleunigung	556
12.4.1	Mnemonics	556
12.4.2	Accelerators	559
13	Prozeßkommunikation	563
13.1	Grundlagen	564
13.2	Atome speichern Namen	564
13.3	Fenster haben Properties	565
13.3.1	Properties lesen	566
13.3.2	Properties erzeugen und verändern	568
13.3.3	Portabilitätsprobleme bei Strukturen	569
13.3.4	Properties löschen	570
13.3.5	Properties und Events	570
13.3.6	Vordefinierte Property-Namen	571
13.4	Kommunizieren mit Events	572
13.4.1	Properties und <i>ClientMessage</i> -Events im Zusammenspiel	573
13.5	Selektionen	582
13.6	Das Motif-Clipboard	583
13.6.1	Die Clipboard-Schnittstelle	583
13.6.2	Daten auf das Clipboard kopieren	586
13.6.3	Daten vom Clipboard lesen	589
13.6.4	Verschiedene Formate	593
13.6.5	Große Datenmengen	593
13.7	Kommunikation mit dem Window-Manager	593
13.7.1	Properties für den Window-Manager	594
13.7.2	Ein Property für Protokolle	595

13.7.3	Window-Manager-Protokolle am Beispiel	595
14	Spezielle Callbacks	599
14.1	Callback-Routinen mit Verzögerung	599
14.2	Callbacks an File-Descriptoren	602
14.3	Callbacks für die Wartezeiten	604
C	Programmierung von Widget-Klassen	605
15	Klassengesellschaft	607
15.1	Die Klassenhierarchie von Motif	608
15.2	Der Aufbau einer Widget-Klasse	610
16	Die Basisklassen der Toolkit-Intrinsics	615
16.1	Die Klasse <i>Core</i>	615
16.1.1	Der Klassen-Record von <i>Core</i>	615
16.1.2	Der Instanzen-Record von <i>Core</i>	618
16.2	Die Klasse <i>Composite</i>	620
16.2.1	Der Klassen-Record von <i>Composite</i>	620
16.2.2	Der Instanzen-Record von <i>Composite</i>	621
16.3	Die Klasse <i>Constraint</i>	622
16.3.1	Der Klassen-Record von <i>Constraint</i>	622
16.3.2	Der Instanzen-Record von <i>Constraint</i>	623
16.4	Abarbeitung der Felder des Klassen-Records	623
16.5	Das Ressourcen-Array eines Widgets	624
16.6	Die Funktionen einer Widget-Klasse	626
16.6.1	Die Funktion <i>initialize()</i>	626
16.6.2	Die Funktion <i>resize()</i>	627
16.6.3	Die Funktion <i>expose()</i>	628
16.6.4	Die Funktion <i>set_values()</i>	628
16.6.5	Die Funktion <i>destroy()</i>	629
16.7	Klassen- und Instanzenzeiger	629
17	Die Klasse <i>XmIconButton</i>	631
17.1	<i>IconB.h</i> – das öffentliche Include-File	634
17.2	<i>IconBP.h</i> – das private Include-File	636
17.3	<i>IconB.c</i> – das Source-File	638
17.3.1	Das Ressourcen-Array	639
17.3.2	Initialisierung des Klassen-Records	641
17.3.3	Funktionen des Icon-Buttons	644

Literaturverzeichnis	653
Anhang: Quelltexte	655
A.1 Textausgaben in Callback-Routinen mit langer Ausführungszeit	655
A.2 Benutzungsoberfläche für Bourne-Shells	657
Sachverzeichnis	667