

Contents

Preface.....	xxvii
Acknowledgments.....	xxix
Introduction.....	xxxi
Authors.....	xxxvii

PART I User Interaction

Chapter 1	Object-Oriented Analysis and Design.....	3
1.1	Introduction	3
1.2	Background Research.....	4
1.3	Software Requirements	4
1.4	General Use Case Narratives.....	5
1.5	Use Case Scenarios and Conversations	5
1.5.1	Scenario.....	6
1.5.2	Conversation.....	8
1.6	Noun and Verb Analysis.....	8
1.7	Object Analysis.....	8
1.8	Preliminary Class Diagram.....	9
1.9	Basic Program Implementing the Class Structure	13
1.10	Summary	15
	References	15
Chapter 2	Initial Graphical User Interface	17
2.1	Introduction	17
2.2	Application Wizard	17
2.3	Menus	17
2.4	Application Icon	20
2.5	Toolbars	21
2.5.1	Inserting a Common Operations Toolbar.....	21
2.5.2	Inserting a Common Blocks Toolbar	22
2.5.3	Attaching the Toolbars to the Application Frame.....	23
2.5.4	Controlling the Toolbar Visibility	25
2.5.5	Add Event-Handler Functions for the Menu Entries.....	27
2.5.6	Add Event-Handler Functions for the Common Blocks Toolbar Buttons.....	29
2.6	Block Library Dialog Window	30
2.6.1	Insert a New Dialog Resource.....	30
2.6.2	Attach a Class to the Dialog Window	31
2.6.3	Menu Entry and Toolbar Button Event-Handling Functions	33
2.6.4	Attach Variables to the Controls	33

2.6.5	Add Functionality for the Block Library Dialog Buttons	34
2.6.5.1	Disable Duplicate Check Boxes.....	34
2.6.5.2	Add All Blocks	35
2.6.5.3	Uncheck All Blocks	40
2.7	Add Functionality to the Common Blocks Toolbar Buttons	41
2.8	Summary	41
	References	41
Chapter 3	Constructing Blocks	43
3.1	Introduction	43
3.2	Merging the WIN32 Console Application with the Visual C++ Application.....	43
3.3	Modifying Program Structure for Block Construction	46
3.3.1	Preliminary Changes.....	46
3.3.1.1	Class CSystemModel.....	46
3.3.1.2	Block Construction Code.....	47
3.3.1.3	Class CBlock.....	48
3.3.2	Introduce a New CBlock Constructor	50
3.3.3	Add a Constructed Block to the Block List.....	51
3.3.4	Check Object Construction	53
3.4	Drawing of Primitive Block Shapes	54
3.4.1	General Drawing Functionality.....	54
3.4.2	Drawing an Ellipse Block	56
3.4.3	Drawing a Rectangle Block.....	57
3.4.4	Drawing a Triangle Block	58
3.4.5	Automatic Assignment of Block Position.....	60
3.5	Completing Block Construction	62
3.5.1	Construction of a Gain Block.....	62
3.5.2	Construction of a Sum Block	63
3.5.3	Construction of All Remaining Blocks.....	64
3.5.3.1	Construction of a Derivative Block	64
3.5.3.2	Construction of a Divide Block	65
3.5.3.3	Construction of an Integrator Block	66
3.5.3.4	Construction of a Linear Function Block	66
3.5.3.5	Construction of an Output Block.....	67
3.5.3.6	Construction of a Signal Generator Block.....	68
3.5.3.7	Construction of a Subsystem Block	68
3.5.3.8	Construction of a Subsystem In Block.....	69
3.5.3.9	Construction of a Subsystem Out Block.....	70
3.5.3.10	Construction of a Transfer Function Block.....	70
3.5.3.11	Check Construction of System Model.....	72
3.6	Drawing Block Graphical Detail.....	73
3.6.1	Implementing Functionality to Draw Detail	73
3.6.1.1	CConstantBlock::DrawBlock()	74
3.6.1.2	CGainBlock::DrawBlock()	75
3.6.1.3	CSumBlock::DrawBlock()	76
3.6.1.4	CDerivativeBlock::DrawBlock()	77
3.6.1.5	CDivideBlock::DrawBlock()	79
3.6.1.6	CIntegratorBlock::DrawBlock()	81
3.6.1.7	CLinearFnBlock::DrawBlock()	83
3.6.1.8	COutputBlock::DrawBlock()	84

3.6.1.9	CSignalGeneratorBlock::DrawBlock()	86
3.6.1.10	CSubsystemBlock::DrawBlock()	88
3.6.1.11	CSubsystemInBlock::DrawBlock()	89
3.6.1.12	CSubsystemOutBlock::DrawBlock()	92
3.6.1.13	CTransferFnBlock::DrawBlock()	94
3.6.2	Display of Block Graphics	96
3.7	Summary	96
	References	96
Chapter 4	Constructing Block Ports	97
4.1	Introduction	97
4.2	Block-Based Port Position	97
4.2.1	Ports on an Ellipse Block	97
4.2.2	Ports on a Rectangle Block	98
4.2.3	Ports on a Triangle Block	99
4.3	Changing the CPort Class	102
4.3.1	CPort Member Variables	102
4.3.2	CPort Member Methods	103
4.3.2.1	Accessor Methods	103
4.3.2.2	Calculate Port Position	104
4.3.2.3	CPort Constructor	108
4.4	Augmenting the CBlock Class	108
4.5	Augmenting the CBlockShape Class	109
4.6	Derived-Block Constructors	110
4.6.1	CConstantBlock Constructor	110
4.6.2	Check the System Model	111
4.6.3	Augmenting the Remaining Block Constructors	112
4.7	Drawing Ports	113
4.7.1	Call CBlock::DrawBlockPorts()	113
4.7.2	Add CBlock::DrawBlockPorts()	114
4.7.3	Add CPort::DrawPort()	114
4.7.4	Display of Block Ports	117
4.8	Summary	118
Chapter 5	Constructing Connections	121
5.1	Introduction	121
5.2	Adding Connections to the DiagramEng Project	121
5.2.1	Augment the CConnection Class	121
5.2.2	Add a Function to Add the Connection to the List	122
5.2.3	Add a Function to Get the Connection List	122
5.2.4	Add an OnLButtonDown() Function	123
5.2.5	Add an OnMouseMove() Function	123
5.2.6	Add an OnLButtonUp() Function	124
5.2.7	Draw Connections	125
5.2.8	Build and Run the Project	129
5.3	Context Menu	130
5.3.1	Insert a Context Menu	130
5.3.2	Add an Event-Handler Function for the Context Menu	130
5.3.3	Set the Context Menu Point	131

5.3.4	Attach Event-Handler Functions for the Context Menu Entries	131
5.3.4.1	Item Deletion	132
5.3.4.2	Set Item Properties	134
5.3.5	System Model–Based Object Deletion.....	134
5.4	Summary	135
	References	135
Chapter 6	Moving Blocks and Connections	137
6.1	Introduction	137
6.2	Moving Blocks.....	137
6.2.1	CRectTracker Class.....	137
6.2.2	Augmenting the OnLButtonDown() Function.....	137
6.2.3	Moving a Block	139
6.2.4	Set Block Position.....	140
6.3	Moving Connection End Points.....	141
6.3.1	Move a Connection End Point.....	141
6.3.2	Snap Connection End Points to a Port	144
6.3.2.1	Augment TrackConnectionEndPoint()	144
6.3.2.2	Augment SetBlockPosition()	147
6.3.3	Unsnap Connection End Points from a Port.....	149
6.4	Inserting and Moving Connection-Based Bend Points	151
6.4.1	Augment the Context Menu with an Insert Bend Point Operation	151
6.4.2	Drawing a Connection with Head, Tail, and Bend Points	155
6.4.2.1	DrawConnection()	155
6.4.2.2	Amend DrawArrowHead()	156
6.4.2.3	Interaction Problem Concerning Invisible Connections	158
6.4.3	Moving a Connection-Based Bend Point	159
6.5	Deletion of Connections and Bend Points.....	162
6.5.1	Deletion of a Bend Point	162
6.5.2	Deletion of a Connection.....	164
6.6	Summary	167
	References	168
Chapter 7	Automatic Block Placement	169
7.1	Introduction	169
7.2	Hypothetical Interactive Block Positioning.....	169
7.3	Actual Improved Automatic Block Positioning.....	172
7.4	Summary	175
Chapter 8	Connection-Based Bend Points	177
8.1	Introduction	177
8.2	Attaching a Connection Tail Point to a Bend Point.....	177
8.2.1	Amend the TrackConnectionEndPoint() Function.....	179
8.2.2	Add Member Variables and Methods to the CConnection Class.....	180
8.2.3	Add a SnapConnectionTailPointToBendPoint() Function.....	181
8.2.4	Problem Concerning the Connection Tail Point	182

8.3	Deletion of a Bend Point to Which a Connection Is Attached	187
8.4	Deletion of a Connection with a Bend Point to Which a Connection Is Attached.....	189
8.5	Problem Concerning the Insertion of Bend Points.....	191
8.6	Automatic Attachment of Connection End Points upon Initial Construction	196
8.7	Deletion of Blocks with Attached Connections	200
8.8	Summary	202
Chapter 9	Block Dialog Windows.....	203
9.1	Introduction	203
9.2	Processing a Left-Button-Double-Click Event	203
9.3	Adding Dialog Windows to Accept User Input.....	204
9.3.1	Constant Block Dialog	205
9.3.1.1	Insert a New Dialog Window and Add Controls.....	205
9.3.1.2	Attach a Class to the Dialog Window	205
9.3.1.3	Attach Variables to the Dialog Window Controls	206
9.3.1.4	Add Functionality to the Dialog Window Buttons	207
9.3.1.5	Add Functionality to Initialize Variables	208
9.3.1.6	Add the Overriding BlockDlgWndParameterInput() Function.....	208
9.3.1.7	Combined Left-Button-Down and Left-Button-Double-Click Problem	210
9.3.2	Derivative Block Dialog	212
9.3.2.1	Insert a New Dialog Window and Add Controls.....	212
9.3.2.2	Attach a Class to the Dialog Window	212
9.3.2.3	Attach Variables to the Dialog Window Controls	213
9.3.2.4	Add Functionality to the Dialog Window Buttons	213
9.3.2.5	Add Functionality to Initialize Variables	214
9.3.2.6	Add the Overriding BlockDlgWndParameterInput() Function.....	214
9.3.3	Divide Block Dialog.....	214
9.3.3.1	Insert a New Dialog Window and Add Controls.....	215
9.3.3.2	Attach a Class to the Dialog Window	215
9.3.3.3	Attach Variables to the Dialog Window Controls	216
9.3.3.4	Add Functionality to the Dialog Window Buttons	216
9.3.3.5	Add Functionality to Initialize Variables	217
9.3.3.6	Add the Overriding BlockDlgWndParameterInput() Function.....	217
9.3.4	Gain Block Dialog.....	218
9.3.4.1	Insert a New Dialog Window and Add Controls.....	218
9.3.4.2	Attach a Class to the Dialog Window	218
9.3.4.3	Attach Variables to the Dialog Window Controls	218
9.3.4.4	Add Functionality to the Dialog Window Buttons	218
9.3.4.5	Add Functionality to Initialize Variables	220
9.3.4.6	Add the Overriding BlockDlgWndParameterInput() Function.....	221
9.3.5	Integrator Block Dialog.....	221
9.3.5.1	Insert a New Dialog Window and Add Controls.....	221
9.3.5.2	Attach a Class to the Dialog Window	221

9.3.5.3	Attach Variables to the Dialog Window Controls	221
9.3.5.4	Add Functionality to the Dialog Window Buttons	223
9.3.5.5	Add Functionality to Initialize Variables	223
9.3.5.6	Add the Overriding BlockDlgWndParameterInput() Function.....	224
9.3.6	Linear Function Block Dialog.....	224
9.3.6.1	Insert a New Dialog Window and Add Controls.....	224
9.3.6.2	Attach a Class to the Dialog Window	224
9.3.6.3	Attach Variables to the Dialog Window Controls	225
9.3.6.4	Add Functionality to the Dialog Window Buttons	226
9.3.6.5	Add Functionality to Initialize Variables	227
9.3.6.6	Add the Overriding BlockDlgWndParameterInput() Function.....	227
9.3.7	Output Block Dialog.....	228
9.3.7.1	Insert a New Dialog Window and Add Controls.....	228
9.3.7.2	Attach a Class to the Dialog Window	228
9.3.7.3	Attach Variables to the Dialog Window Controls	229
9.3.7.4	Add Functionality to the Dialog Window Buttons	229
9.3.7.5	Add Functionality to Initialize Variables	230
9.3.7.6	Add the Overriding BlockDlgWndParameterInput() Function.....	230
9.3.8	Signal Generator Block Dialog	231
9.3.8.1	Insert a New Dialog Window and Add Controls.....	231
9.3.8.2	Attach a Class to the Dialog Window	231
9.3.8.3	Attach Variables to the Dialog Window Controls	231
9.3.8.4	Add Functionality to the Dialog Window Buttons	233
9.3.8.5	Add Functionality to Initialize Variables	233
9.3.8.6	Add the Overriding BlockDlgWndParameterInput() Function.....	234
9.3.9	Subsystem Block Dialog.....	235
9.3.9.1	Insert a New Dialog Window and Add Controls.....	235
9.3.9.2	Attach a Class to the Dialog Window	235
9.3.9.3	Attach Variables to the Dialog Window Controls	236
9.3.9.4	Add Functionality to the Dialog Window Buttons	236
9.3.9.5	Add Functionality to Initialize Variables	237
9.3.9.6	Add the Overriding BlockDlgWndParameterInput() Function.....	237
9.3.10	Subsystem In Block Dialog	237
9.3.10.1	Insert a New Dialog Window and Add Controls.....	238
9.3.10.2	Attach a Class to the Dialog Window	238
9.3.10.3	Attach Variables to the Dialog Window Controls	238
9.3.10.4	Add Functionality to the Dialog Window Buttons	239
9.3.10.5	Add Functionality to Initialize Variables	239
9.3.10.6	Add the Overriding BlockDlgWndParameterInput() Function.....	240
9.3.11	Subsystem Out Block Dialog.....	240
9.3.11.1	Insert a New Dialog Window and Add Controls.....	240
9.3.11.2	Attach a Class to the Dialog Window	240
9.3.11.3	Attach Variables to the Dialog Window Controls	241
9.3.11.4	Add Functionality to the Dialog Window Buttons	241

9.3.11.5	Add Functionality to Initialize Variables	242
9.3.11.6	Add the Overriding BlockDlgWndParameterInput() Function	242
9.3.11.7	Synchronizing the Subsystem-Related Block Values	243
9.3.12	Sum Block Dialog	243
9.3.12.1	Insert a New Dialog Window and Add Controls	243
9.3.12.2	Attach a Class to the Dialog Window	243
9.3.12.3	Attach Variables to the Dialog Window Controls	244
9.3.12.4	Add Functionality to the Dialog Window Buttons	244
9.3.12.5	Add Functionality to Initialize Variables	245
9.3.12.6	Add the Overriding BlockDlgWndParameterInput() Function	245
9.3.13	Transfer Function Block Dialog	246
9.3.13.1	Insert a New Dialog Window and Add Controls	247
9.3.13.2	Attach a Class to the Dialog Window	247
9.3.13.3	Attach Variables to the Dialog Window Controls	248
9.3.13.4	Add Functionality to the Dialog Window Buttons	248
9.3.13.5	Add Functionality to Initialize Variables	249
9.3.13.6	Add the Overriding BlockDlgWndParameterInput() Function	249
9.4	Summary	250
	References	250
Chapter 10	Conversion of String Input to Double Data	251
10.1	Introduction	251
10.2	Conversion of CString Input Strings to Double Member Data	251
10.2.1	Add a Global ConvertStringToDouble() Function	253
10.2.2	Add a Global StripInputString() Function	256
10.2.3	Add a Global Data-Dimension-Determining Function	258
10.2.4	Add a Global ConvertStringMatrixToDoubleMatrix() Function	261
10.2.5	Add Global Diagnostic Printing Functions	263
10.3	Augmenting the CBlock-Derived Classes	264
10.3.1	Augment the CConstantBlock Class	264
10.3.2	Augment the CGainBlock Class	267
10.3.3	Augment the CIntegratorBlock Class	269
10.3.4	Augment the CTransferFnBlock Class	273
10.4	Summary	277
	Reference	277
Chapter 11	Moving Multiple Items	279
11.1	Introduction	279
11.2	Edit the TrackItem() Function to Call TrackMultipleItems()	279
11.3	Add a TrackMultipleItems() Function to CDiagramEngDoc	280
11.3.1	Add the DetermineCurrentAndIntersectRects() Function	285
11.3.2	Add the DetermineTrackerDeltaPosition() Function	286
11.4	Add a Function for a Key-Down Event to CDiagramEngView	286
11.5	Add Key-Flag-Based Members to CDiagramEngDoc	287

11.6	Flow of Control in the <code>TrackMultipleItems()</code> Function	289
11.7	Add a Toolbar Button to Activate Multiple Item Tracking.....	290
11.8	Summary	291
	Reference.....	291
Chapter 12	Addition of a Tree View Control.....	293
12.1	Introduction	293
12.2	Add a Tree View Control Dialog Window	293
12.2.1	Insert a Dialog Window and Add Controls	293
12.2.2	Attach a Class to the Dialog Window	293
12.2.3	Add Functionality to Invoke the Tree View Dialog	294
12.2.4	Attach a Variable to the Dialog Window Control.....	296
12.2.5	Add the <code>OnInitDialog()</code> Function to the CTreeDialog Class.....	297
12.2.6	Add Icons to the Tree View Control	299
12.2.7	Add an Event-Handler Function for Tree View Node Selection	304
12.3	Docking of the Tree Dialog Window as a Tree Dialog Bar	307
12.3.1	Step 1: Change the Base Class from <code>CDialog</code> to <code>CDialogBar</code>	308
12.3.2	Step 2: Change the Constructor and <code>DoDataExchange()</code> Methods.....	308
12.3.3	Step 3: Add a New Declaration for the <code>OnInitDialog()</code> Function with Different Parameters.....	309
12.3.4	Step 4: Alter the Message Map to Invoke the <code>OnInitDialog()</code> Function.....	311
12.3.5	Step 5: Add the Implementation of the New <code>OnInitDialog()</code> Function.....	311
12.3.6	Step 6: Alter the Dialog Box Resource Styles.....	312
12.3.7	Step 7: Add an Instance of the <code>CDialogBar</code> to the <code>CMainFrame</code> Class	312
12.3.8	Step 8: Invoke the <code>Create()</code> Method for the New <code>CDialogBar</code> Instance	313
12.3.9	Step 9: Add Support for Dynamic Docking and Resizing of the <code>CDialogBar</code> Instance.....	316
12.4	Resetting the Appropriate Dialog.....	316
12.5	Controlling the Toolbar Visibility	317
12.6	Summary	319
	References	319
Chapter 13	Review of Menu and Toolbar-Based Functionality: Part I.....	321
13.1	Introduction	321
13.2	Menus	321
13.2.1	Main Frame–Based Menus	321
13.2.2	Child Frame–Based Menus.....	321
13.2.3	Context Menu	321
13.3	Toolbars	325
13.3.1	Main Frame Toolbar.....	325
13.3.2	Common Operations Toolbar.....	326
13.3.3	Common Blocks Toolbar.....	327

13.4	Functionality to be Added	327
13.4.1	Child Frame–Based Menu Items	327
13.4.2	Context Menu Items	329
13.4.3	Toolbar Button–Based Functionality	329
13.5	Summary	330
Chapter 14	Context Menu Extension	331
14.1	Introduction	331
14.2	Numerical Solver Dialog Window	331
14.2.1	Insert a New Dialog Window and Add Controls	331
14.2.2	Attach a Class to the Dialog Window	333
14.2.3	Attach Variables to the Dialog Window Controls.....	333
14.2.4	Add Functionality to the Dialog Window Buttons.....	333
14.2.5	Add Functionality to Initialize Variables.....	334
14.2.6	Add a Dialog Window Parameter Input Function to the Underlying Class	335
14.2.7	Call the Dialog Window Parameter Input Function.....	337
14.3	Deleting Multiple Grouped Items.....	337
14.3.1	Change the Name of the Edit Menu Entry	338
14.3.2	Add a Delete Grouped Items Entry to the Context Menu	338
14.3.3	Add an Event-Handler Function to Delete Grouped Items	338
14.4	Setting of Item Properties.....	343
14.5	Summary	345
	Reference.....	345
Chapter 15	Setting Port Properties	347
15.1	Introduction	347
15.2	Setting Divide Block Ports	347
15.2.1	Reset Input Ports	347
15.2.2	Calculate Port Position Angles.....	351
15.2.3	Reset Output Ports.....	353
15.3	Setting Sum Block Ports.....	355
15.3.1	Reset Input Ports	355
15.3.2	Calculate Port Position Angles.....	358
15.3.3	Reset Output Ports.....	361
15.4	Drawing Ports Depending on Connection Status.....	363
15.5	Drawing Port Signs.....	365
15.6	Preparation for Adjustment of Port Properties	373
15.7	Dialog-Based Port Property Setting.....	377
15.7.1	Insert a Dialog Window and Add Controls.....	378
15.7.2	Attach a Class to the Dialog Window	378
15.7.3	Attach Variables to the Dialog Window Controls.....	378
15.7.4	Add Functionality to the Dialog Window Buttons.....	383
15.7.5	Add Overriding Versions of the SetPortProperties() Function.....	383
15.8	Deleting Ports	390
15.9	Summary	397

Chapter 16	Key-Based Item Movement	399
16.1	Introduction	399
16.2	Add a Fine Move Item Entry to the Context Menu	399
16.3	Add Members to the CDiagramEngDoc Class	400
16.4	Edit the CDiagramEngView::OnKeyDown() Function	401
16.5	Add a FineMoveItem() Function	402
16.6	Add a FineMoveBlock() Function	402
16.7	Add a FineMoveConnectionBendPoint() Function	406
16.8	Fine Movement Process Control Flow	409
16.9	Summary	411
	Reference	411
Chapter 17	Reversing Block Direction	413
17.1	Introduction	413
17.2	Add a Reverse Block Entry to the Context Menu	413
17.3	Add a ReverseBlockDirection() Function to CBlock Class	415
17.4	Add Accessor Methods to the CBlockShape Class	416
17.5	Add Accessor Methods to the CPort Class	417
17.6	Test the Application	417
17.7	Summary	418
 PART II Model Computation and Data		
Chapter 18	Model Validation	421
18.1	Introduction	421
18.2	Validating a Model	421
18.2.1	Validate Model Blocks	424
18.2.2	Validate Model Block Ports	426
18.2.3	Validate Model Connections	431
18.3	Changes to be Made to Draw in the Correct Color	435
18.3.1	Modify the CSystemModel::DrawSystemModel() Function	436
18.3.2	Modify the CConnection::DrawConnection() Function	437
18.3.3	Modify the CConnection::DrawArrowHead() Function	438
18.4	Unique Port and Bend Point Connection	441
18.4.1	Augment the SnapConnectionHeadPointToPort() Function	441
18.4.2	Augment the SnapConnectionTailPointToPort() Function	443
18.5	Summary	446
	Reference	446
Chapter 19	Non-Feedback-Based Signal Propagation	447
19.1	Introduction	447
19.2	Signals and Systems	447
19.2.1	Continuous-Time and Discrete-Time Signals	447
19.2.2	Continuous-Time and Discrete-Time Systems	447

19.2.3	Memory-Based Systems and Memoryless Systems	448
19.2.4	Time Variant and Time Invariant Systems	448
19.2.5	Linear and Nonlinear Systems	449
19.2.6	State Variables, State Vectors, and State Space	449
19.3	Models with Algebraic and Feedback Loops	449
19.4	Signal Propagation.....	451
19.5	Determination of Feedback Loops	452
19.5.1	General Steps to Determine Feedback Loops.....	452
19.5.1.1	Determine Model Feedback Loops	454
19.5.1.2	Set Reference-From Port of Connection Emanating from a Bend Point	457
19.5.1.3	Assign Block Connectivity Matrix Values	459
19.5.1.4	Determine Loops from Connectivity Matrix	460
19.5.2	Determining Loops from the Connectivity Matrix.....	462
19.5.2.1	Determine Initial Node Vector	462
19.5.2.2	Build Tour	463
19.5.2.3	Find Current Tour Position	467
19.5.2.4	Find Branch Nodes	467
19.5.2.5	Number of Ones on Matrix Row	468
19.5.2.6	Check Repeated Entries.....	468
19.5.2.7	Reset Tour Vector	469
19.5.2.8	Save Tour Vector.....	469
19.5.2.9	Print Integer Matrix	470
19.5.2.10	Print Integer Vector	471
19.5.2.11	Running the Application.....	472
19.6	Signal-Based Class Structure	473
19.7	Additions Required for Direct Signal Propagation	474
19.7.1	Addition of Signal Propagation Code.....	474
19.7.1.1	Add Code to the CDiagramEngDoc::OnSimStart() Function	475
19.7.1.2	Modify the CSystemModel::ValidateModel() Function	476
19.7.1.3	Modify the CDiagramEngDoc::OnModelBuild() Function	476
19.7.1.4	Add a CSystemModel::GetNFeedbackLoops() Function	477
19.7.1.5	Add a CSystemModel::SignalPropagation Direct() Function	477
19.7.1.6	Add a CSystemModel::DetermineNTime Steps() Function	481
19.7.1.7	Add a CSignal::ResetSignalData() Function	482
19.7.1.8	Add a CSystemModel::SignalPropagation SystemOfEquations() Function	483
19.7.1.9	Add a Virtual CBlock::OperateOnData() Function	483
19.8	Propagating a Simple Direct Signal	484
19.8.1	Implementation of Simple Signal Propagation.....	485
19.8.1.1	Add a CConstantBlock::OperateOnData() Function	485
19.8.1.2	Add a COutputBlock::OperateOnData() Function	485

19.8.1.3	Add a CBlock::WriteDataToOutputSignal() Function	489
19.8.1.4	Add a CBlock::ReadDataFromSingleInput Signal() Function	490
19.8.1.5	Add Accessor Methods to Get and Set Signal Data and Validity.....	492
19.8.2	Running an Experiment	494
19.9	Summary	495
	References	495
Chapter 20	Graph Drawing.....	497
20.1	Introduction	497
20.2	Addition of Structure to Display an Empty View Window.....	497
20.2.1	Add a Show Graph Button to the Output Block Dialog	497
20.2.2	Add a CDocTemplate Object and Accessor Function to the CDiagramEngApp Class.....	499
20.2.3	Modify CDiagramEngApp::InitInstance() to Create a CMultiDocTemplate Object.....	499
20.2.4	Add a COutputBlockView Class to Support Drawing within Its View	501
20.2.5	Add Template-Based Functionality to the Show Graph Event-Handler Function	503
20.2.6	Add the OnInitialUpdate() and OnUpdate() Functions to the CDiagramEngView Class	505
20.3	Supplementing Existing Classes to Access Output Block Data	506
20.3.1	Add a Member Variable to the COutputBlockView Class	506
20.3.2	Modify COutputBlock::BlockDlgWndParameter Input()	506
20.3.3	Add a Member Variable to the COutputBlockDialog Class	507
20.3.4	Modify the COutputBlockDialog Constructor Declaration and Definition	508
20.3.5	Add Code to COutputBlockDialog::OnOutputBlkDlg BtnShowGraph()	508
20.3.6	Add Accessor Functions to the COutputBlockView Class	510
20.4	Plotting Data as a Text String.....	510
20.4.1	Add a COutputBlock::GetDataDimensions() Function.....	510
20.4.2	Extend the COutputBlockView::OnDraw() Function	511
20.5	Plotting Numerical Data as a Graph.....	512
20.5.1	Modify OnDraw() in Preparation for Plotting Curves.....	512
20.5.2	Add a COutputBlockView::DetermineMaxMin Values() Function.....	514
20.5.3	Add DrawGraph() to the COutputBlockView Class.....	515
20.5.4	Drawing Problem due to a Change in Active Document Status	519
20.5.5	Add SelectGraphCurveColor() to the COutputBlockView Class.....	520
20.5.6	Add GetTimeParameters() to the CSystemModel Class.....	521
20.6	Deleting an Output Block and Its View.....	522
20.6.1	Delete the Output Block View.....	523
20.6.2	Modify CDiagramEngDoc::DeleteBlock()	524
20.6.3	Modify CDiagramEngDoc::OnEditDeleteGroupedItems()	525

20.7	Adding Functionality to the Output Block Dialog Window.....	526
20.7.1	Add Accessor Functions to the COutputBlock Class.....	527
20.7.2	Amend COutputBlockView::DrawGraph()	527
20.7.3	Amend COutputBlock::BlockDlgWndParameterInput()	529
20.8	Summary	531
	References	531
Chapter 21	Block Operations.....	533
21.1	Introduction	533
21.2	Linear Function Block Data Operation	533
21.3	Signal Generator Block Data Operation.....	535
21.4	Gain Block Data Operation	538
21.5	Sum Block Data Operation.....	544
21.6	Derivative Block Data Operation	549
21.6.1	Three-Point Derivative.....	549
21.6.2	Five-Point Derivative.....	550
21.6.3	Implementing the OperateOnData() Function.....	551
21.6.3.1	Addition of an OperateOnData() Function	551
21.6.3.2	Addition of the CalculateThreePointDerivative() Function....	557
21.6.3.3	Addition of the CalculateFivePointDerivative() Function.....	557
21.6.3.4	Addition of Class Member Variables and Their Initialization.....	558
21.6.3.5	Addition of a ResetBlockData() Function	559
21.6.3.6	Addition of a ResetMemoryBasedBlockData() Function	560
21.6.3.7	Testing the Derivative Block.....	561
21.7	Integrator Block Data Operation	563
21.7.1	Integration	563
21.7.2	Truncation Error and Method Order	565
21.7.3	Higher-Order Methods	566
21.7.4	Implementing an Integration Function.....	566
21.7.5	Implementing the OperateOnData() Function.....	568
21.7.5.1	Addition of an OperateOnData() Function	568
21.7.5.2	Addition of the EulerMethod() Function.....	574
21.7.5.3	Addition of the GetIntegrationMethod() Function....	574
21.7.5.4	Addition of Class Member Variables and Their Initialization.....	575
21.7.5.5	Addition of a ResetBlockData() Function	575
21.7.6	Running an Experiment	577
21.8	Divide Block Data Operation	577
21.8.1	Implementing the OperateOnData() Function.....	578
21.8.2	Implementing the DivideMultiplyOperation() Function	581
21.8.3	Implementing the GaussJordanEliminationFullPivoting() Function	599
21.8.4	Implementing the Swap() Function.....	604
21.9	Remaining Block Operation Functions	605
21.10	Summary	605
	References	606

Chapter 22	Preparation for Feedback-Based Signal Propagation	607
22.1	Introduction	607
22.2	Linear Systems	607
22.3	Nonlinear Systems.....	608
22.3.1	Newton's Method.....	609
22.3.2	Computing a Linear System with a Nonlinear Method	610
22.4	Model Assumptions.....	611
22.4.1	State-Space Equations.....	611
22.4.2	Resorting of the Block List	613
22.4.3	Saving Tour Vectors	616
22.4.4	Get a Saved Tour Vector from a List.....	619
22.4.5	Building a Block List from an Integer Array	620
22.5	Determining Initial Output Signals.....	622
22.5.1	Check Signal Validity.....	622
22.5.2	Preliminary Signal Propagation for Models with Feedback Loops	623
22.6	Adding Functionality to Set the Initial Output Signal	629
22.6.1	Adding a Dialog Window to Allow User Input.....	629
22.6.1.1	Insert a Dialog Window and Add Controls	630
22.6.1.2	Attach a Class to the Dialog Window	630
22.6.1.3	Attach Variables to the Dialog Window Controls	630
22.6.1.4	Add Functionality to the Dialog Window Buttons	631
22.6.1.5	Add Functionality to Initialize Variables	631
22.6.2	Adding a Context Menu and Updating User Input.....	632
22.6.2.1	Add a Context Menu Entry to Set the Initial Output Signal	633
22.6.2.2	Add an Event-Handler Function Named OnSetOutputSignal()	633
22.6.2.3	Add an AssignBlockOutputSignal() Function to Set the Signal Value.....	634
22.7	Building the System of Equations Representing the Block Diagram	639
22.7.1	Order of Operations to Compute a Model with Feedback Loops.....	640
22.7.1.1	Preliminary Signal Propagation	645
22.7.1.2	Data Dimensions.....	645
22.7.1.3	Construction of System Signal Vector.....	646
22.7.1.4	Construction of System Function Operation Vector.....	646
22.7.1.5	Newton Solver.....	646
22.7.1.6	Signal Vector Assignment	646
22.7.1.7	Execution of Output Blocks.....	646
22.7.1.8	Resetting of States	647
22.7.2	Additional Functions Required to Compute a Model with Feedback Loops	647
22.7.2.1	Determining the System Equations Data Dimensions	647
22.7.2.2	Determining the System Equations Signal Vector	650
22.7.2.3	Determining the System Equations Function Operation Vector.....	652
22.7.2.4	Newton Method Iterative Solver	655
22.7.2.5	Updating of the System Equations Signal Vector.....	655
22.7.3	Testing the Code Structure	658

22.8	Stopping a Simulation	658
22.8.1	Augment the CSystemModel::SignalPropagation Direct() Method.....	660
22.8.2	Augment CSystemModel::SignalPropagationSystem OfEquations()	661
22.8.3	Add a CheckStopButtonPressed() Method to the CSystemModel Class	662
22.8.4	Complete the CDiagramEngDoc::OnSimStop() Event-Handler Function.....	664
22.9	Summary	665
	References	666
Chapter 23	Feedback-Based Signal Propagation	667
23.1	Introduction	667
23.2	Newton-Method-Based Solver.....	667
23.3	Object-Oriented Transformation of the Newton-Method-Based Solver	669
23.3.1	Header File Used in the Object-Oriented Approach.....	670
23.3.2	Source File Used in the Object-Oriented Approach	672
23.3.3	Test Case Problems Computed Using Newton's Method.....	687
23.3.3.1	Case 1.....	687
23.3.3.2	Case 2.....	688
23.3.3.3	Case 3.....	688
23.3.3.4	Case 4.....	688
23.3.3.5	Case 5.....	688
23.4	Adding the Newton-Method-Based Solver to the DiagramEng Project	688
23.4.1	Addition of Details to Complete the NewtonSolver() Function	689
23.4.2	Addition of the CNewtonMethod Class with Member Declarations	690
23.4.3	Addition of the CNewtonMethod Class Member Function Definitions	692
23.4.4	Changes to be Made to Resolve Exiting the Program Prematurely	697
23.4.4.1	Modify nrerror()	697
23.4.4.2	Modify lnsrch()	698
23.4.4.3	Modify ludcmp()	699
23.4.4.4	Modify newt()	700
23.4.4.5	Modify All nrerror() Calls Concerning Memory Allocation Failure.....	703
23.4.4.6	Modify Relevant Functions to Process the Integer Returned from newt()	704
23.4.5	Addition of the Manual Assertion Functions to the Project.....	705
23.5	Convergence of the Newton Method Used to Compute Models with Feedback Loops.....	708
23.6	Testing the Newton-Method-Based Code for Models with Feedback Loops	710
23.6.1	Computing a Linear Problem with a Feedback Loop	710
23.6.2	Computing an Ordinary Differential Equation Using a Feedback Loop.....	711
23.6.3	Computing the Linear State Equations	714
23.6.3.1	Mechanical System.....	715
23.6.3.2	Electrical System	720
23.6.4	Computing a Coupled Linear System	721

23.6.5	Computing the Lotka–Volterra Equations	723
23.6.6	Computing the Lorenz Equations	728
23.7	Summary	732
	References	732
Chapter 24	Placing an Edit Box on a Toolbar	733
24.1	Introduction	733
24.2	Edit the Resource File	733
24.3	Create the Toolbar Edit Box	734
24.4	Update the Toolbar Edit Box	736
24.5	A Memory Leak	740
24.6	Summary	741
	References	741
Chapter 25	Serialization	743
25.1	Introduction	743
25.2	System Model Data Structure	743
25.2.1	CSystemModel Data	744
25.2.2	CBlock Data	744
25.2.3	CBlockShape Data	745
25.2.4	CPort Data	745
25.2.5	CConnection Data	746
25.2.6	CSignal Data	746
25.2.7	Derived-Block Class Data	747
25.2.7.1	CConstantBlock Data	747
25.2.7.2	CDerivativeBlock Data	747
25.2.7.3	CDivideBlock Data	747
25.2.7.4	CGainBlock Data	747
25.2.7.5	CIntegrator Data	748
25.2.7.6	CLinearFnBlock Data	748
25.2.7.7	COutputBlock Data	749
25.2.7.8	CSignalGeneratorBlock Data	749
25.2.7.9	CSubsystemBlock Data	749
25.2.7.10	CSubsystemInBlock Data	749
25.2.7.11	CSubsystemOutBlock Data	750
25.2.7.12	CSumBlock Data	750
25.2.7.13	CTransferFnBlock Data	751
25.3	Event-Handler Functions for File Input/Output	751
25.3.1	Child Frame–Based Event-Handler Functions	751
25.3.1.1	CDiagramEngDoc::OnFileOpen()	752
25.3.1.2	CDiagramEngDoc::ReadDiagramEngData FromFile()	755
25.3.1.3	CDiagramEngDoc::OnFileSaveAs()	755
25.3.1.4	CDiagramEngDoc::OnFileSave()	756
25.3.1.5	CDiagramEngDoc::OnFileClose()	757
25.3.1.6	CDiagramEngApp::OnAppExit()	758
25.3.1.7	CDiagramEngApp::SaveAllDocuments()	760
25.3.1.8	CDiagramEngApp::OnFileRecentFileOpen() ...	763
25.3.1.9	CDiagramEngDoc::OnWndCloseAllDocs()	767

25.3.2	Main Frame–Based Event-Handler Functions	768
25.3.2.1	CDiagramEngApp::OnFileOpen()	769
25.4	Writing Data Members to a File	770
25.4.1	Writing CDiagramEngDoc Data to a File.....	770
25.4.2	Writing CSystemModel Data to a File.....	773
25.4.3	Writing CBlock Data to a File.....	775
25.4.4	Writing CBlockShape Data to a File.....	776
25.4.5	Writing CPort Data to a File	776
25.4.6	Writing Derived-Block Class Data to a File.....	777
25.4.6.1	CConstantBlock::WriteBlockDataToFile()	777
25.4.6.2	CDerivativeBlock::WriteBlockDataToFile()	777
25.4.6.3	CDivideBlock::WriteBlockDataToFile()	778
25.4.6.4	CGainBlock::WriteBlockDataToFile()	778
25.4.6.5	CIntegratorBlock::WriteBlockDataToFile()	779
25.4.6.6	CLinearFnBlock::WriteBlockDataToFile()	779
25.4.6.7	COutputBlock::WriteBlockDataToFile()	780
25.4.6.8	CSignalGeneratorBlock::WriteBlockData ToFile()	780
25.4.6.9	CSubsystemBlock::WriteBlockDataToFile() ...	781
25.4.6.10	CSubsystemInBlock::WriteBlockData ToFile()	781
25.4.6.11	CSubsystemOutBlock::WriteBlockData ToFile()	782
25.4.6.12	CSumBlock::WriteBlockDataToFile()	782
25.4.6.13	CTransferFnBlock::WriteBlockData ToFile()	783
25.4.7	Writing CConnection Data to a File.....	783
25.4.8	Example of File Output	784
25.5	Reading Data Members from a File	787
25.5.1	Reading CSystemModel Data from a File	788
25.5.2	Reading CConnection Data from a File.....	793
25.5.3	Reading CBlock Data from a File	794
25.5.4	Reading CBlockShape Data from a File	795
25.5.5	Reading CPort Data from a File.....	796
25.5.6	Reading Derived-Block Class Data from a File.....	796
25.5.6.1	CConstantBlock::ReadBlockDataFrom File()	796
25.5.6.2	CDerivativeBlock::ReadBlockDataFrom File()	798
25.5.6.3	CDivideBlock::ReadBlockDataFromFile()	799
25.5.6.4	CGainBlock::ReadBlockDataFromFile()	799
25.5.6.5	CIntegratorBlock::ReadBlockDataFrom File()	800
25.5.6.6	CLinearFnBlock::ReadBlockDataFromFile()	802
25.5.6.7	COutputBlock::ReadBlockDataFromFile()	802
25.5.6.8	CSignalGeneratorBlock::ReadBlockData FromFile()	803
25.5.6.9	CSubsystemBlock::ReadBlockDataFrom File()	804
25.5.6.10	CSubsystemInBlock::ReadBlockDataFrom File()	805

25.5.6.11	CSubsystemOutBlock::ReadBlockData FromFile()	805
25.5.6.12	CSumBlock::ReadBlockDataFromFile()	806
25.5.6.13	CTransferFnBlock::ReadBlockDataFrom File()	806
25.6	Saving the Initial Output Signal for the Divide Block	807
25.6.1	Add Member Variables to the CDivideBlock Class	808
25.6.2	Amend the CDivideBlock Constructor and Destructor Functions.....	808
25.6.3	Add the Overriding AssignBlockOutputSignal() Function to the CDivideBlock Class	809
25.6.4	Amend the CDivideBlock Class Serialization Functions	813
25.6.5	Amend the PreliminarySignalPropagation() Function	814
25.7	Adding Data-Writing Functionality to the Output Block	819
25.7.1	Add a Save Data Button to the Output Block Dialog Resource	819
25.7.2	Add an Event-Handler Function for the Save Data Button	819
25.8	Summary	824
	References	825

PART III Refinement

Chapter 26	Review of Menu and Toolbar-Based Functionality: Part II	829
26.1	Introduction	829
26.2	Functionality to Be Added	829
26.3	Summary	830
Chapter 27	Printing and Print Preview	831
27.1	Introduction	831
27.2	Additions to the DiagramEng Project	832
27.3	Mapping Modes.....	833
27.3.1	Coordinate Axes.....	834
27.3.2	Scale of Transformation	835
27.4	Augmenting the CDiagramEngView OnBeginPrinting() Function.....	836
27.5	Augmenting the CDiagramEngView OnPrepareDC() Function.....	840
27.6	Augmenting the CDiagramEngView OnDraw() Function.....	842
27.7	Summary	844
	References	844
Chapter 28	Implementing a Scroll View.....	845
28.1	Introduction	845
28.2	Coordinate Systems	845
28.3	Conversion from CView to CScrollView	847
28.3.1	Changes to "DiagramEngView.h"	847
28.3.2	Changes to "DiagramEngView.cpp"	847
28.3.3	Changes to CDiagramEngView::OnInitialUpdate()	851
28.4	Device Point to Logical Point Coordinate Conversions	852
28.4.1	Amending the CDiagramEngView-Based Functions	852
28.4.1.1	CDiagramEngView::OnContextMenu()	852
28.4.1.2	CDiagramEngView::OnLButtonDownDb1Clk()	853

28.4.1.3	CDiagramEngView::OnLButtonDown()	854
28.4.1.4	CDiagramEngView::OnLButtonUp()	855
28.4.1.5	CDiagramEngView::OnMouseMove()	856
28.4.2	Amending the CDiagramEngDoc-Based Functions	857
28.4.2.1	CDiagramEngDoc::TrackBlock()	857
28.4.2.2	CDiagramEngDoc::TrackConnectionBend Point()	859
28.4.2.3	CDiagramEngDoc::TrackConnectionEnd Point()	860
28.5	Tracking of Multiple Items	864
28.6	Limiting Diagram Entity Placement	870
28.7	Fitting the Logical View to the Physical View	874
28.8	Zooming In and Out of the View	877
28.8.1	Addition of Detail to the CDiagramEngView Zooming Functions	877
28.8.2	Addition of a ScaleSystemModel() Function to the CSystemModel Class	879
28.8.3	Resetting the Default Model Geometry	880
28.9	Summary	884
	References	884

Chapter 29 Edit Menu 885

29.1	Introduction	885
29.2	Selection of All Content	885
29.2.1	Add a Select-All-Related Member Variable to the CDiagramEngDoc Class	885
29.2.2	Complete the CDiagramEngDoc::OnEditSelectAll() Function	886
29.2.3	Add a Function to the CSystemModel Class to Get the Diagram Coordinate Minima	888
29.2.4	Add a Resetting Mechanism to Cancel the Select-All Operation	890
29.3	Cut, Copy, and Paste	893
29.3.1	Add CDiagramEngDoc-Based Event-Handler Functions	894
29.3.2	Add the CClipboard Class Declaration and Definition	899
29.3.3	Add Class Copy Constructors and Related Methods	901
29.3.3.1	CBlock Copy Constructor	901
29.3.3.2	CBlockShape Copy Constructor	903
29.3.3.3	CPort Copy Constructor	903
29.3.3.4	CConnection Copy Constructor	904
29.3.3.5	CConstantBlock Copy Constructor	904
29.3.3.6	CDerivativeBlock Copy Constructor	907
29.3.3.7	CDivideBlock Copy Constructor	908
29.3.3.8	CGainBlock Copy Constructor	909
29.3.3.9	CIntegratorBlock Copy Constructor	911
29.3.3.10	CLinearFnBlock Copy Constructor	912
29.3.3.11	COutputBlock Copy Constructor	913
29.3.3.12	CSignalGeneratorBlock Copy Constructor	915
29.3.3.13	CSubsystemBlock Copy Constructor	916
29.3.3.14	CSubsystemInBlock Copy Constructor	917
29.3.3.15	CSubsystemOutBlock Copy Constructor	918

29.3.3.16	CSumBlock Copy Constructor	919
29.3.3.17	CTransferFnBlock Copy Constructor	920
29.3.4	Testing the Cut, Copy, and Paste Operations for a Complete Model	922
29.4	Undoing and Redoing Editing Actions	922
29.4.1	Add the Undo and Redo Event-Handler Functions to the CDiagramEngDoc Class	924
29.4.2	Add Functionality to Save System Models to a List to the CDiagramEngDoc Class	929
29.4.3	Add a Copy Constructor to the CSystemModel Class	931
29.4.4	Add an Assignment Operator to the CSystemModel Class	935
29.4.5	Add Functionality to Update the User Interface to the CDiagramEngView Class	937
29.4.6	Testing the Editing Actions	939
29.5	Summary	941
	References	941
Chapter 30	Annotations	943
30.1	Introduction	943
30.2	Add a CAnnotation Class to the Project	943
30.3	Add an Annotation Dialog Window to the Project	946
30.3.1	Add the Format Annotation Entry to the Context Menu	946
30.3.2	Add the New Dialog Window and Associated Functionality	950
30.3.2.1	Insert a New Dialog Window and Add All Necessary Controls	950
30.3.2.2	Attach a Class to the Dialog Window	952
30.3.2.3	Attach Variables to the Dialog Window Controls	952
30.3.2.4	Attach Functionality to Display a Font List	953
30.3.2.5	Add Functionality to the Dialog Window Buttons	958
30.4	Add an Annotation to a System Model	960
30.5	Deleting Annotations	963
30.6	Moving Annotations	966
30.6.1	Fine Movement of an Annotation	966
30.6.2	Direct Movement of an Annotation	970
30.6.3	Multiple Annotation Movement	974
30.7	View Menu–Based Actions	976
30.7.1	Augmenting Auto Fit Diagram	976
30.7.2	Zooming In and Out of a Diagram	979
30.7.3	Resetting Initial Diagram Geometry	980
30.8	Format Menu–Based Actions	980
30.8.1	Augment Document–Based Classes	980
30.8.2	Updating the User Interface	983
30.9	Edit Menu–Based Actions	984
30.9.1	Cut, Copy, and Paste Actions Involving Annotations	985
30.9.1.1	Add a Copy Constructor and Relevant Methods to the CAnnotation Class	985
30.9.1.2	Add CAnnotation Specific Variables and Methods to the CClipboard Class	986
30.9.1.3	Augment the OnEditCut(), OnEditCopy(), and OnEditPaste() Functions	987

30.9.2	Undo and Redo of Editing Actions Involving Annotations	991
30.9.2.1	Augment the Model Saving Function	991
30.9.2.2	Augment the CSystemModel Copy Constructor	992
30.9.2.3	Augment the CSystemModel Assignment Operator	993
30.10	Serialization of the CAnnotation Class	994
30.10.1	CAnnotation Class Serialization	995
30.10.1.1	CAnnotation::WriteAnnotationData ToFile()	995
30.10.1.2	CAnnotation::ReadAnnotationData FromFile()	996
30.10.2	CSystemModel Class Serialization	998
30.10.3	Augmenting the File Open Methods	1001
30.11	Summary	1003
References		1004
Chapter 31	Tools Menu	1005
31.1	Introduction	1005
31.2	Tools Menu Implementation: Diagnostic Information	1005
31.2.1	Presentation Space Application Programming Interface	1005
31.2.2	Structures, Functions, and Fields Required for Memory-Related Information.....	1006
31.2.3	Add a Dialog Window to Display Diagnostic Memory-Related Information.....	1007
31.2.3.1	Insert a New Dialog Window and Add All Necessary Controls.....	1007
31.2.3.2	Attach a Class to the Dialog Window	1007
31.2.3.3	Attach Variables to the Dialog Window Controls	1007
31.2.3.4	Add Functionality to the Dialog Window Buttons	1007
31.2.3.5	Add Functionality to Initialize Variables	1011
31.2.3.6	Create the Dialog Window	1011
31.3	Summary	1014
References		1015
Chapter 32	Help Menu	1017
32.1	Introduction	1017
32.2	Using DiagramEng.....	1017
32.3	Creating the Using DiagramEng Document	1017
32.3.1	CreateProcess() Function	1018
32.3.2	OnHelpUsingDiagramEng() Function	1018
32.4	Summary	1022
References		1023
Chapter 33	Finalizing the Project	1025
33.1	Introduction	1025
33.2	Preventing Usage of Nonfunctional Blocks.....	1025
33.3	Preventing the Building of Subsystems.....	1027
33.4	Disabling Nonfunctional Items	1027

33.5	Preparing the Source and Executable Files.....	1028
33.5.1	Memory Leak Detection	1028
33.5.2	Release Version of the Software.....	1029
33.5.3	Source Code	1030
33.6	Summary	1031
	References	1031
Chapter 34	Conclusion	1033
34.1	Introduction	1033
34.2	Project Review.....	1033
34.2.1	User Interaction	1033
34.2.2	Model Computation and Data	1034
34.2.3	Refinement.....	1035
34.3	Improvements	1036
34.3.1	Object-Oriented Analysis and Design.....	1036
34.3.1.1	Class Hierarchy.....	1036
34.3.1.2	Retrieving the Address of the Document	1037
34.3.2	Addition of Classes.....	1037
34.3.3	Unnecessary Features.....	1037
34.3.4	Naming Convention.....	1038
34.3.5	Member Variables and Constant Accessor Methods	1038
34.3.6	Program State.....	1039
34.3.7	Generic Containers.....	1039
34.3.8	Base Class Selection.....	1039
34.3.9	Serialization	1040
34.3.10	Activity Organization.....	1040
34.4	Summary	1041
	References	1042
Appendix A: ControlEng: Win32 Console Application	1043	
Appendix B: Constructing Connections: An Exploration	1063	
Appendix C: NodeArcConnectivity: Win32 Console Application	1077	
Appendix D: Debugging: An Introduction.....	1093	
Appendix E: MatrixInversion: Win32 Console Application	1115	
Appendix F: Using DiagramEng	1133	
Index.....	1151	