2 C++ Overview 6

2.1 Objects and Classes 7

2.2 Encapsulation 7

2.3 Modularity and Scoping 8

2.4 Inheritance 8

2.5 Method Overloading 8

2.6 Exceptions 9

2.7 Hello World Example 9

3 Overview of C++ ORBs 11

3.1 Terminology 11

3.2 Clients and Servers as C++ Applications 12

3.3 Clients and Servers Implemented with Non-C++ ORBs 15

4 Building a First C++ ORB Application 17

4.1 Summary of the CORBA Development Process 18

4.2 Environment Setup 19

4.3 Interface Specification 19

4.4 Compiling the IDL 20

4.5 A Client as a C++ Application 21

4.6 Object Implementation 23

4.7 A Server as a C++ Application 24

4.8 Compiling the Server and the Client 26

4.9 Running the Application 27

Chapter 2
CORBA Overview 29

1 The Object Management Group 30

1 The Object Management Group 30

1.1 OMG's Goals 30

1.2 The Organizational Structure of the OMG 30

1.3 OMG Technology Adoption Process 34

2 The Object Management Architecture 35

2.1 Overview of the OMA 36

2.2 Core Object Model 36

2.3 The Reference Model 41

3 Common Object Request Broker Architecture 44

3.1 Overview 44

3.2 Object Model 46

3.3 ORB Structure 49

3.4 OMG Interface Definition Language (IDL) 52

3.5 ORB and Object Interfaces 63

3.6 Basic Object Adapter 71

3.7 The Portable Object Adapter 75

3.8 Language Mappings 84

3.9 Interoperability 85

3.10 TypeCode, Any, and DynAny 87

3.11 Dynamic Invocation and Dynamic Skeleton Interfaces 92

3.12 Interface Repository 94

5.12 Interface Repository 94

Chapter 3
OMG IDL to C++ Mapping 107

1 Mapping Modules 107

2 Mapping Basic DataTypes 108

3 Mapping Strings 109

4 Mapping Constants 111

5 Mapping Enumerations 112

6 Mapping for Structured Types 112

6.1 Mapping for Struct Types 114

6.2 Mapping Union Types 116

6.3 Mapping Sequence Types 118

6.4 Mapping for Arrays 121

6.5 Mapping Typedefs 123

6.6 Mapping the Type Any 124

7 Mapping for the Exception Types 130

8 Mapping Operations and Attributes 131

9 Argument Passing 133

9.1 Examples 136

10 Mapping of Interfaces 163

10.1 Client-Side Mapping 163

10.2 Server-Side Mapping 163

Chapter 4 167
ORB Runtime System

1 Object Interface 168

1.1 get_implementation() 168

1.2 get_interface() 168

1.3 is_nil() 169

1.4 duplicate() and release() 169

1.5 is_a() 169

1.6 non_existent() 169

1.7 is_equivalent() 170

1.8 hash() 170

1.9 create_request() 170

2 ORB Interface 171

2.1 ORB Initialization 171

2.2 Converting Object References into Strings and Vice Versa 171

2.3 Obtaining Initial References 172

2.4 BOA Initialization 173

2.5 POA Initialization 173

3 Basic Object Adapter 173

3.1 Activation and Deactivation 173

3.2 Other Operations 174

4 TypeCodes 175

4.1 Interface TypeCode 175

4.2 Creating TypeCodes 179

4.2 Creating TypeCodes 179

5 Dynamic Invocation Interface 182

5.1 Common Data Structures 182

5.2 Creating an NVList 183

5.3 NVList Interface 183

5.4 DII Request 184

5.5 Creating a Request 184

5.6 Request Interface 185

6 Dynamic Skeleton Interface 187

6.1 ServerRequest Interface 188

7 Context Interface 188

7.1 Creating a Context Object 189

7.2 Manipulating a Context Object 189

7.3 Manipulating the Context Object Tree 190

8 Portable Object Adapter 191

8.1 POA Policies 191

8.2 POAManager Interface 195

8.3 AdapterActivator Interface 197

8.4 ServantActivator Interface 198

8.5 ServantLocator Interface 199

8.6 POA Interface 201

8.7 Current Operations 207

Chapter 5
Discovering Services 209

1 Bootstrapping 209

2 The CORBA Naming Service 210

2.1 Overview of the Naming Service 210

2.2 Interface Specification 211

2.3 Using the Naming Service from a C++ Client 218

3 Trading Service 221

3.1 Overview of Trading 222

3.2 Overview of the Trading Service Interfaces 226

3.3 Exporting a Service Offer 238

4 Domains 257

5 Proprietary Object Location 258

Chapter 6
Building Applications 261

1 Application Specification 261

1.1 IDL Specification 263

2 Implementing Objects 264

2.1 Implementing the Meeting Object 264

2.2 Implementing the Room Object 265

3 Building Servers 268

3.1 Initializing the ORB 270

3.2 Creating an Object, Registering with the Root POA 270

3.3 Registering with the Naming Service 271

3.3 Registering with the Naming Service 271

3.4 Entering the ORB's Event Loop 272

4 Building Factories 272

4.1 Meeting Factory Object Implementation 273

4.2 Meeting Factory Server 274

5 Starting Servers 276

6 Building Clients 277

6.1 Client Application 278

6.2 Methods in the Client Application 279

7 Extensions to the Example Application 287

Chapter 7
Advanced Features 289

1 The Extended Hello World Example 289

1.1 Interface Specification 289

1.2 A Client 290

1.3 Servant Implementation 292

2 The Any Type and TypeCodes 294

2.1 Interface Specification 294

2.2 Object Implementation 295

2.3 Client Implementation 297

3 Interface Repository and Dynamic Invocation Interface 301

3.1 Initializing the ORB 302

3.1 Initializing the ORB 302

3.2 Browsing the Interface Repository 304

3.3 A Simple Unparser 305

3.4 Initializing Requests 307

3.5 Creating Supporting Objects 307

3.6 Using the Supporting Objects 309

3.7 Creating and Invoking a Request Object 310

3.8 Getting Results 310

3.9 Executing the Client 312

4 Dynamic Skeleton Interface 314

5 Tie Mechanism 317

6 IDL Context 321

6.1 Creating a Context 321

6.2 Invoking the Method 322

6.3 Getting Values from the Context 322

Glossary 325

Index 331

Foreword

While it may seem somewhat retrograde in 1999 to publish a book which centers on C++ (rather than the hot C-based language, Java), this book will find an important place in the library of programmers everywhere. Even as Java turns 35 in dog years (oops, I guess I mean Internet years!), or about five in human reckoning, according to most surveyors of the programming scene C++ is still the primary C-based language, especially in mission-critical, high-performance systems. Fortunately, the CORBA technology discussed in this tome (as was

3.2 Browsing the Interface Repository 304

3.3 A Simple Unparser 305

3.4 Initializing Requests 307

3.5 Creating Supporting Objects 307

3.6 Using the Supporting Objects 309

3.7 Creating and Invoking a Request Object 310

3.8 Getting Results 310

3.9 Executing the Client 312

4 Dynamic Skeleton Interface 314

5 Tie Mechanism 317

6 IDL Context 321

6.1 Creating a Context 321

6.2 Invoking the Method 322

6.3 Getting Values from the Context 322

Glossary 325

Index 331

Foreword

While it may seem somewhat retrograde in 1999 to publish a book which centers on C++ (rather than the hot C-based language, Java), this book will find an important place in the library of programmers everywhere. Even as Java turns 35 in dog years (oops, I guess I mean Internet years!), or about five in human reckoning, according to most surveyors of the programming scene C++ is still the primary C-based language, especially in mission-critical, high-performance systems. Fortunately, the CORBA technology discussed in this tome (as was