

Indice

Prefazione	XV
Capitolo 1. Il <i>middleware</i>	1
1.1 Introduzione	1
1.2 L'integrazione di applicazioni	1
1.3 Il <i>middleware</i>	4
1.3.1 Caratteristiche generali	4
1.3.2 Modelli di <i>middleware</i>	6
1.4 Il modello di chiamata di procedura remota	8
1.5 Il modello orientato ai messaggi	15
1.6 Il modello transazionale	17
1.7 Il modello dello spazio delle tuple	19
1.8 Il modello di accesso remoto ai dati	22
1.9 Il modello a oggetti distribuiti	24
1.10 Il modello a componenti	27
Capitolo 2. CORBA e OMA	33
2.1 Introduzione	33
2.2 L'architettura OMA	34
2.3 Il modello a oggetti CORBA	37
2.4 L'interoperabilità tra ORB	40
2.4.1 Generalità	40
2.4.2 La <i>Interoperability Architecture</i>	41
2.5 Il modello a componenti CORBA	43
2.5.1 Caratteristiche fondamentali	43
2.5.2 Modello astratto	46

Capitolo 3. Ciclo di sviluppo di applicazioni CORBA	49
3.1 Introduzione	49
3.2 Struttura dell' <i>Object Request Broker</i>	49
3.2.1 La <i>Interface Repository</i>	52
3.2.2 Il <i>Portable Object Adapter</i>	53
3.3 Ciclo di sviluppo	53
3.4 Esempio	55
3.4.1 Definizione dell'interfaccia IDL	55
3.4.2 Compilazione dell'interfaccia IDL	55
3.4.3 Sviluppo dell'applicazione cliente	57
3.4.4 Sviluppo dell'oggetto servente	59
3.4.5 Distribuzione e esecuzione	61
Capitolo 4. Il linguaggio IDL	63
4.1 Introduzione	63
4.2 Regole e convenzioni lessicali	66
4.3 Direttive di precompilazione	67
4.4 Interfacce, attributi e operazioni	67
4.5 Moduli	70
4.6 Tipi di dato	71
4.6.1 Tipi primitivi	71
4.6.2 Dichiarazioni <code>typedef</code>	73
4.6.3 Tipi enumerativi	73
4.6.4 Tipi strutturati	73
4.6.5 Tipi nativi	75
4.7 Costanti	76
4.8 Eccezioni	76
4.9 Ereditarietà tra interfacce	79
4.10 Contesti	81
4.11 Valuetype	82
Capitolo 5. Da IDL a Java	83
5.1 Introduzione	83
5.2 Identificatori riservati	84
5.3 Classi <i>Holder</i>	85
5.4 Classi <i>Helper</i>	85
5.5 Interfacce, attributi e operazioni	85
5.6 Moduli	86
5.7 Tipi di dato	87
5.7.1 Tipi primitivi	87
5.7.2 Tipi enumerativi	88

5.7.3 Tipi strutturati	88
5.8 Costanti	89
5.9 Eccezioni	89
5.10 Contesti	89
5.11 Valuetype	90
Capitolo 6. CORBA dal lato del cliente	93
6.1 Introduzione	93
6.2 Le interfacce SII e DII	93
6.3 SII - Modello concettuale	94
6.4 SII - Modello di programmazione	95
6.5 SII - Esempi di programmazione	96
6.5.1 Operazione con parametro di ingresso di tipo semplice	97
6.5.2 Operazione con parametro di ingresso-uscita di tipo semplice	98
6.5.3 Operazione con parametro di ingresso-uscita di tipo d'utente	99
6.5.4 Operazione con valore di ritorno di tipo d'utente	100
6.5.5 Operazione con eccezione definita dall'utente	101
6.5.6 Operazione con valore di ritorno di tipo <code>valuetype</code>	102
6.5.7 Operazione con valore di ritorno di tipo <code>any</code>	103
6.6 Classi di libreria	104
6.6.1 <code>org.omg.CORBA.Object</code>	104
6.6.2 <code>org.omg.CORBA.Request</code>	105
6.6.3 <code>org.omg.CORBA.Any</code>	107
6.6.4 <code>org.omg.CORBA.portable.ObjectImpl</code>	109
6.6.5 <code>org.omg.CORBA.portable.OutputStream</code> e <code>InputStream</code>	109
6.6.6 <code>org.omg.CORBA.ORB</code>	111
6.7 Approfondimenti	112
6.7.1 Problematiche di portabilità degli <i>stub</i>	112
6.7.2 Dentro lo <i>stub</i>	113
6.8 L'interfaccia DII	116
6.8.1 <i>Request</i> per un metodo semplice	117
6.8.2 <i>Request</i> per un metodo che restituisce un valore di tipo semplice	118
6.8.3 <i>Request</i> per un metodo con parametro di scambio	120
6.8.4 La <i>Interface Repository</i>	121
Capitolo 7. CORBA dal lato del servente	123
7.1 Introduzione	123
7.2 Stati di un oggetto CORBA	123
7.3 CORBA dal lato servente	125
7.4 Realizzazione della classe servente	126
7.5 Esempi	127
7.5.1 Operazione con parametro di ingresso di tipo semplice	128

7.5.2	Operazione con parametro di ingresso-uscita di tipo semplice	128
7.5.3	Operazione con parametro di ingresso-uscita di tipo d'utente	129
7.5.4	Operazione con valore di ritorno di tipo d'utente	130
7.5.5	Operazione con eccezione definita dall'utente	131
7.5.6	Operazione con valore di ritorno di tipo <code>valuetype</code>	132
7.5.7	Operazione con valore di ritorno di tipo <code>any</code>	133
7.6	<i>Skeleton</i> e serventi	134
7.6.1	<i>Skeleton</i> come estensione di <code>DynamicImplementation</code>	135
7.6.2	<i>Skeleton</i> come implementazione di <code>InvokeHandler</code>	136
7.6.3	Esempio di <i>skeleton</i> dinamico	137
7.6.4	La classe <code>Servant</code>	138
7.7	Il <i>Portable Object Adapter</i>	140
7.7.1	Le politiche di un POA	142
7.7.2	La creazione e la configurazione del POA	149
7.8	Controllo delle richieste: il <i>POAManager</i>	151
7.9	Il <i>Servant Manager</i>	153
7.9.1	<i>ServantActivator</i>	154
7.9.2	<i>ServantLocator</i>	156
7.10	Il <i>Default Servant</i>	159
7.11	Applicazione servente	161
7.12	Considerazioni riassuntive	162
7.13	Esempi	163
7.13.1	Implementazione delle interfacce	164
7.13.2	Realizzazione di un'applicazione cliente	171
7.13.3	Realizzazione di un'applicazione servente	174
Capitolo 8. I servizi Naming e Trading		189
8.1	Il servizio dei nomi	189
8.1.1	Generalità	189
8.1.2	Struttura del servizio dei nomi	191
8.1.3	Specifiche IDL del servizio dei nomi	192
8.1.4	<code>org.omg.CosNaming.NameComponent</code>	193
8.1.5	<code>org.omg.CosNaming.NamingContext</code>	194
8.2	Il servizio <i>Trading</i>	200
8.2.1	Generalità	200
8.2.2	<i>ServiceTypeRepository</i>	201
8.2.3	<i>Register</i>	201
8.2.4	<i>Lookup</i>	202
Capitolo 9. La comunicazione asincrona		203
9.1	Introduzione	203
9.2	<i>Distributed callback</i>	203

9.2.1	Esempio	204
9.2.2	Limiti del <i>distributed callback</i>	205
9.3	Il servizio di gestione eventi	205
9.3.1	Modello concettuale	205
9.3.2	L'interfaccia dell' <i>Event Service</i>	208
9.3.3	Esempio di programmazione	212
9.3.4	<i>Event Channel Federation</i>	218
9.3.5	Limiti del servizio a eventi	219
9.4	Il servizio di notifica di eventi	220
9.4.1	L'architettura del <i>Notification Service</i>	221
Capitolo 10. I servizi avanzati		225
10.1	Introduzione	225
10.2	Il <i>Licensing Service</i>	225
10.3	Il <i>LifeCycle Service</i>	227
10.4	Il <i>Transaction Service</i>	227
10.5	Il <i>Concurrency Control Service</i>	228
10.6	Il <i>Persistent State Service</i>	229
10.7	Il <i>Security Service</i>	230
10.7.1	La sicurezza nei sistemi distribuiti	230
10.7.2	Modelli per la gestione della sicurezza	231
10.7.3	Descrizione del <i>CORBA Security Service</i>	232
10.7.4	Politiche di sicurezza	234
10.7.5	Meccanismi di sicurezza	235
10.7.6	Livelli di sicurezza	238
Capitolo 11. Piattaforme CORBA commerciali		239
11.1	BEA Web Logic Enterprise(WLE)	239
11.1.1	Generalità	239
11.1.2	Caratteristiche fondamentali	240
11.1.3	Tipologie di clienti	241
11.1.4	Gli <i>Environmental Objects</i>	242
11.1.5	Il <i>Bootstrap Object</i>	243
11.1.6	Il <i>Listener / Handler IIOP</i>	244
11.1.7	Il <i>TP Framework</i>	245
11.1.8	Gestione dello stato di un oggetto	245
11.1.9	Gestione dei guasti	246
11.1.10	Bilanciamento del carico	248
11.1.11	Strumenti di gestione	248
11.2	Borland Enterprise Server VisiBroker	249
11.2.1	Generalità	249
11.2.2	Interoperabilità tra applicazioni in Java e C++	249

11.2.3 Architettura dello <i>Smart Agent</i>	250
11.2.4 Implementazione e attivazione degli oggetti	251
11.2.5 Gestione dei <i>thread</i> e delle connessioni	252
11.2.6 Gestione delle transazioni	253
11.2.7 Autenticazione e crittografia	253
11.2.8 Gestione della sicurezza	254
11.2.9 L' <i>Object Activation Daemon</i>	254
11.2.10 Servizi standard CORBA in VisiBroker	255
11.2.11 VisiBroker per OS/390	255
11.3 IONA Orbacus	256
11.3.1 Generalità	256
11.3.2 La <i>Implementation Repository</i>	257
11.3.3 <i>Portable Interceptors</i>	258
11.3.4 Bilanciamento del carico	258
11.3.5 Ulteriori caratteristiche presenti in ORBACUS	259
11.3.6 ORBACUS Watson	259
11.3.7 Gestione della concorrenza	260
11.3.8 Estensioni per la tolleranza ai guasti	261
11.4 ACE e TAO	261
11.4.1 Generalità	261
11.4.2 Applicazioni CORBA <i>real-time</i> e TAO	261
11.4.3 Portabilità dell'ORB TAO	262
11.4.4 Componenti dell'architettura TAO	263
Capitolo 12. Studio di un caso	265
12.1 Introduzione	265
12.2 Analisi	266
12.2.1 Diagramma di collaborazione	267
12.2.2 Classi	268
12.3 Progettazione	270
12.3.1 Diagramma di distribuzione	270
12.3.2 Diagramma delle classi	270
12.3.3 Interfacce IDL	271
12.3.4 Dinamica inter-oggetti	275
12.4 Programmazione	275
12.4.1 Classe <i>ProdottoImpl</i>	275
12.4.2 Classe <i>DefaultProdottoServant</i>	278
12.4.3 Classe <i>GestoreProdottiImpl</i>	279
12.4.4 Classe <i>GestoreAcquistiImpl</i>	280
12.4.5 Classe <i>RedattoreCatalogoImpl</i>	282
12.4.6 Classe <i>ServerGestoreProdotti</i>	283
12.4.7 Classe <i>ServerRedattoreCatalogo</i>	289
12.4.8 Cliente di prova	291

Appendice A. Grammatica del linguaggio IDL	293
Appendice B. La piattaforma Inprise VisiBroker	301
B.1 Sviluppo di applicazioni Java	301
B.2 Avvio e configurazione dello <i>Smart Agent</i>	303
B.3 Attivazione automatica dei <i>server</i>	304
B.4 Impostazioni delle variabili d'ambiente	307
B.5 Controllo dell'output	307
Elenco degli acronimi	309
Bibliografia	311
Indice analitico	315