Index

2PC (Two-Phase Commit Protocol), 124-125
  distributed 2PC
    avoiding, 143-144
      client controlled transactions, 144-145
      implicit application level protocol, 146-147
      server controlled transactions, 145-146
  limitations, 126-127

abstraction
  components, 17
  functional decomposition, 16
access control lists (ACLs), 194
access layers, binding design rules, 80
ACID (atomicity, consistency, isolation, durability), 123
ACID transactions, limitations of, 125-126
ACLs (access control lists), 194
ADA, 17
ADA programming language, 17
adapters, 96
  intermediary services, 78
adding service orientation to project management
    methodologies, 280
additional runtime features, distribution techniques 29
ADO DataSets, 136
ADO DiffGrams, 136
agility, 241-242
  change requests, 89
  IT, 1
application frontends, SOAs, 58-59
application heterogeneity, 21
application landscape, 56
application level protocol, distributed 2PC, 146-147
application servers, 37
applications, 56
  multi-channel applications, 228-229
    fundamental SOA, 229-230
    process-enabled SOAs, 231-234
    service facades, 230-231
architects
  perspective of SOAs, 253
  SOA architects. *See* SOA architects
architectural roadmap, fundamental SOA, 87
architecture, 55-56
  BPM, 108
  CSG, 350
    asynchronous integration with EBI, 352-353
    Bulk Integration Infrastructure, 353
    choreography, 355
    contracts, 354
    management, 355
    repositories, 354
    security, 355
    service interfaces, 354
    synchronous integration with CSIB, 350-352
  Deutsche Post case study, 320-321
  enterprise architecture versus standards, 7-8
  of enterprise software, 4-5
    requirements of, 6-7
  Intelligent Finance, 362, 369-370
  multichannel architecture, 100
architecture boards, 368
  service repositories, 62
architecture roadmaps
  fundamental SOA, 91-92
  networked SOA, 87, 93-98
  process-enabled SOAs, 87, 98-102
asynchronous communication, 38, 350
  coupling, 47
asynchronous integration, with EBI, 352-353
atomicity, 123
auditing, 172
authenticating against SOA, 188
authentication, 187-190
  creating, 191
  and middleware, 190-192
  SOAP, 192-193
authorization, 193-195
  dynamic authorization,195
  static authorization, 194
automated test tools, 301
automicity, consistency, isolation, durability (ACID), 123
availability, 178-181
  CICS, 185
  CORBA, 184
  EJBs, 182-184
  of enterprise software, 24
  in a heterogeneous SOA, 186
  Web Services, 181
  wrapped legacy applications, 185-186
avoiding
  distributed 2PC, 143-144
    client controlled transactions, 144-145
    implicit application level protocols, 146-147
    server controlled transactions, 145-146
  exposing transaction logic to service clients, 145

B2B (business-to-business), 24, 217-220
  location transparency, 218
  security infrastructures, 217
  stateless semantics, 218
B2B integration, 344
Baan, 24
backers, success of SOAs, 265-266
banking engine services, Intelligent Finance, 371
basic layer, 82
basic services, 70, 364
  data-centric services, 70-71
  Intelligent Finance, 371
  logic-centric services, 72-73
BDM (Business Domain Model), 313
  Deutsche Post, 314
beans, stateless session beans, 183
Berkeley, r-tools suite, 19
BII (Bulk Integration Infrastructure), 163
billing, execution containers, 169
binding
  development-time binding, 62
  runtime binding, 63-64
  service binding, 64
binding design rules, access layers, 80
Boehm, Barry; Spiral Model, 279
bonus systems, 270
BookAndBill, 93, 98
booking process, 99-101
bottom-up code generation, 165-166
BPEL4WS (Business Process Execution Language for Web
    Services), 107
BPM (Business Process Management), 103-105, 120, 155
  architecture of, 108
  versus BPMS, 104-105
  combining with SOA and MOA, 155-156
  overview of, 107
    modeling languages, 107-108
  process integrity, 121, 130-131
  process-enabled SOAs, 112
    core business logic versus process control logic, 112-114
    design implications, 114
  The Third Wave, 104
  vision, 109-111
BPML (Business Process Modeling Language), 107
BPMN (Business Process Modeling Notation), 107-108
BPMS (Business Process Management System), 105, 111
  versus BPM, 104-105
  when to choose, 105-107
BPR (Business Process Reengineering), 103
budgets, success of SOAs, 264
Bulk Integration Infrastructure (BII), 163, 350, 353
business computing, 23-24
  SAP, 23
  service-orientation, 15
  Wal-Mart, 23
Business Domain Model (BDM), 313
business exceptions, 119
business functionality, complexity, 241
business impact
  CSG, 342-343

KrafzigIXfinal.fm Page 376 Friday, October 15, 2004 4:31 PM

Deutsche Post, 313-315
Intelligent Finance, 360-361
Winterthur, 326
business infrastructure, motivation for creating SOAs, 246-247
business level, cost savings, 243
business logic, SOA, 60
Business Process Execution Language for Web Services (BPEL4WS), 107
Business Process Management System. *See* BPMS
Business Process Management. *See* BPM
Business Process Modeling Language (BPML), 107
Business Process Modeling Notation (BPMN), 107-108
Business Process Reengineering (BPR), 103
business processes, complexity, 241
business projects versus IT projects, 281
business rules, 113
business services versus SOA infrastructure, 281
business whitepapers, 266
business-to-business. *See* B2B

callbacks, 190
and polling services, 40
and queues, 42
Carr, Nicolas G., 1
case studies
Credit Suisse Group. *See* CSG
Deutsche Post. *See* Deutsche Post
HBoS. *See* HBoS
Winterthur. *See* Winterthur
central repositories, 269
centralized banking engine service, 364
CEO, perspective of SOAs, 252
Champy, James, 103
change, IT's ability to change, 1
change requests, agility, 89
characteristics of enterprise software, 3-4
choosing
BPMS, 105-107
granularity for transactional steps, 151-153
choreography, CSG, 355
CICS (Customer Information Control Systems), 21, 36, 184-185
CICS log manager, 178
CICS Transaction Gateway (CTG), 185
CIO
conflicts of interests, 258
perspective of SOAs, 252
classification, 69-70
Class-Responsibility-Collaboration (CRC), 279
client controlled transactions, 144-145
clients, fat clients, 221-222
clustering, 20
CM (configuration management), 298
COBOL (Common Business Oriented language), functional decomposition, 16
co-browsing, 233
CODASYL (Conference on Data Systems Languages), 16
code generation, 164
bottom-up approach, 165-166
top-down approach, 165-166
with MDA, 167-168
combining
SOA, MOA, and BPM, 155-156
transaction chains with compensating transactions, 130
Commodore PET, 23
Common Business Oriented Language. *See* COBOL
Common Object Request Broker Architecture. *See* CORBA
communication
asynchronous communication. *See* asynchronous communication
communication middleware. *See* communication middleware framework
minimizing resources for communication, 224
simulated synchronous communication, 39
synchronous communication. *See* synchronous communication
communication middleware framework, 30
application servers, 37
Distributed Objects, 32-33
MOM, 34-35
email, 34
RPCs, 31-32
transaction monitors, 36
communication modes, 28
compensating logic, 153-154
compensating transactions, combining with transaction chains, 130
compensation transactions, 129
complexity, 241-242
component programming, 17
components, 17
concurrency control
optimisitc concurrency control, 135-136
examples, 137-138
pessimistic concurrency control, 135
examples, 139
Conference on Data Systems Languages (CODASYL), 16
configuration management, 296
challenges for, 296-298
recommendations for the SOA integration team, 298-300
configurations
logging, 177-178
runtime configurations, 178
confirm itinerary, 133
conflicts of interest, stakeholders, 258-261
consistency, 123
consolidated logs, 122
containers, 168
contracts, 269
CSG, 354
Deutsche Post case study, 322
Intelligent Finance, 370
service contracts. *See* service contracts
SOA, 59-60
Winterthur, 336-338
CORBA (Common Object Request Broker Architecture), 8, 20
availability, 184
scalability, 184
Winterthur, 338
CORBA (OMG), 160
CORBA IDL, 20
core business logic, 113
cost effectiveness, service, 14
cost savings, 242-243
at business level, 243
IT, 243-244
coupling, 47
loose coupling versus tight coupling, 46-49
CRC (Class-Responsibility-Collaboration), 279
create invoice, 133
Credit Suisse Group. *See* CSG
Credit Suisse Information Bus (CSIB), 344
CRM (Customer Relationship Management), 4, 15
cross-container integration, 170-171
CSG (Credit Suisse Group), 341-342
implementing SOA, 346
processes and structures, 347-348
project management, 349
repositories, 348
lessons learned from implementing SOA, 355-356
project scope, 342
business impact, 342-343
technology impact, 343-346
technology, 350
architecture, 350-355
CSIB (Credit Suisse Information), 163, 344
synchronous integration 350, 351, 352
CTG (CICS Transaction Gateway), 185
cursor stability, 125
Customer Information Control Systems. *See* CICS
Customer Relationship Management (CRM), 4, 15, 24
customer retention service, 303

Dahl, Ole-Johan, 17
data, SOA, 60
data access services, 113
data integrity, 117
versus process integrity, 117
user-defined data integrity, 118
databases, 23
data-centric services, 70-71
DCE (Distributed Computing Environment), 20, 32
debugging, 172
decomposition, 242
decoupling
of functionality and technology, requirements of enterprise software architecture, 7
public enterprise services, 81
from technology, 242
design, BPM and process-enabled SOAs, 114
Design in Action (DIA), 367
designing
authentication, 191

KrafzigIXfinal.fm Page 377 Friday, October 15, 2004 2:08 PM

for small devices, 223-227
Deutsche Post, 311-312
BDM, 314
implementing SOA, 316
processes and structures, 316-318
project management, 319
service registry, 318-319
project scope, 312-313
business impact, 313-315
technology impact, 315
results of implementing SOA, 323-324
SBB, 315
SSB, 322
technology, 320-322
Deutsche Post World net, 311
development processes, motivation for creating SOAs, 247
development-time binding, 62
DIA (Design in Action), 367
discontinuous networks, limitations of 2PC, 127
dispatching execution containers, 168
distributed 2PC, 124
avoiding, 143-144
client controlled transactions, 144-145
implicit application level protocol, 146-147
server controlled transactions, 145-146
distributed computing, 19-22
SOAP, 22
XML, 21
Distributed Computing Environment (DCE), 20, 32
distributed logging, 173-175
Distributed Objects, 32-33
distribution techniques, heterogeneity, 27-29
additional runtime features, 29
communication modes, 28
products, 29
distribution technology, service-orientation, 15
divide and conquer strategies, 290, 292
documentation, 242
service documentation, 250
document-centric messages, 45-46
domain, 117
domain inconsistencies, 119
domain-specific business services, 327
drivers, test drivers, 306
durability, 123
dX method, 279

EAI (Enterprise Application Integration), 21, 24, 211-212
service enablement, 212-215
service repositories, 217
service stability, 215-216
SOAs, 215
upgrade ability, 215-216
EBI (Event Bus Infrastructure), 163, 350
asynchronous integration, 352-353
eCommerce, 24
EDMs (Enterprise Data Models), 7
EJB (Enterprise Java Beans)
availability, 182-184
containers, 168
scalability, 182-184
email, 34
embedded messages. *See* payload semantics
encapsulation, components, 17
Encina, 36
encryption, 196-197
enterprise applicatioin integration, 92
Enterprise Application Integration. *See* EAI
Enterprise Data Models (EDMs), 7
enterprise IT renovation roadmaps, 3, 10-11, 262
Enterprise Java Beans. *See* EJB
enterprise layer, 82
Enterprise Resource Planning (ERP), 15, 24, 211
Enterprise Service Bus, 160
enterprise software
architecture versus standards, 7-8
architecture of, 4-5
requirements of, 6-7
availability of, 24
characteristics of, 3-4
Enterprise Software Bus, 8
enterprise software systems, lack of agility and inefficiency, 2-3
enterprise-level software project management, 284
entity, 117
entity relationship models (ER), 71
entropy, 4
e-Platform, 325, 331
ER (entity relationship models), 71
ERP (Enterprise Resource Planning), 15, 24, 211
error handling, idempotent operations, 143
error reporting, 172-173
establishing project management methodologies, 278-280
SOA-driven project management, 281-282
Event Bus Infrastructure (EBI), 163, 350
evolution, motivation for creating SOAs, 248-249
example scenarios, travel itinerary management, 133-134
examples
of optimistic concurrency control, 137-138
of pessimistic concurrency control, 139
of scenarios, passenger check-in scenario, 205-207
exceptions, out of stock exceptions, 119
execution containers, 168
cross-container integration, 170-171
logging, 169
message transformation, 169
security, 168
expansion stages, 88
fundamental SOA, 91-92
networked SOA, 93-98
process-enabled SOAs, 98-102
exposing transaction logic to service clients, 145
Extreme Programming, 279

facades, intermediary services, 78-80
failures, 171
auditing, 172
distributed logging, 173-175
error reporting, 172-173
example of SOA failure, 271-272
fatal failures, 141
in individual process steps, 129
logging. *See* logging
fat clients, 221-222
fatal failures, 141
feedback, motivation for creating SOAs, 249
File Transfer Protocol (FTP), 39
fine-grained interaction patterns, stateful session beans, 183
Fingar, Peter, 104
fire-and-forget RPC, 39
flexibility, requirements of enterprise software architecture, 6
frameworks
logging, 177-178
sign-on frameworks, 193
FTP (File Transfer Protocol), 39
functional decomposition, 16
functional departments, perspective of SOAs, 254
functional programming, 16
functional testing, 301
functionality-adding services, intermediary services, 81
fundamental SOA, 87, 91-92
multi-channel applications, 229-230

GIS (geographic information systems), 4
goals of SOA, 10
granularity, 242
choosing for transactional steps, 151-153
software artifacts, 284
Greenfield Project, Intelligent Finance, 361

Halifax Bank of Scotland (HBoS), 359-360
Hammer, Michael, 103
hardware buses, 159. *See also* software buses
HBoS (Halifax Bank of Scotland), 359
Intelligent Finance. *See* Intelligent Finance
heterogeneity
distribution techniques, 27-29
and security, 199-200
horizontal slicing versus vertical slicing, 287
hub-and-spoke, 110

IAD (Iterative Application Development), 288
IBM, MQSeries, 160, 185
IBM PC, 23
IBM WebSphere MQ, 34
ideal worlds
SOA specifics, 269-271
structures and processes, 266-269
idempotent, 206
idempotent update operations, 140-142
error handling, 143
sequence numbers, 142
IDL (Interface Definition Language), 20
IF.com, Intelligent Finance, 361
IIS (Internet Information Server), 37

KrafzigIXfinal.fm Page 378 Friday, October 15, 2004 2:08 PM

implementation, SOA, 60
implementing
SOA at CSG, 346
processes and structures, 347-348
project management, 349
repositories, 348
SOA at Deutsche Post, 316
processes and structures, 316-318
project management, 319
service registry, 318-319
SOA at Intelligent Finance, 365
project management, 367-369
repositories, 366
XML, 365-366
SOA at Winterthur, 330
processes and structures, 330-332
project management, 333-334
repositories, 332-333
IMS (Information Management System), 36
inconsistencies
domain inconsistencies, 119
process inconsistencies, 118
independence from technology, 245-246
Information Management System (IMS), 36
integration, 343-344
asynchronous integration with EBI, 352-353
complexity, 242
of legacy systems and packaged applications, 126
of purchased software, 344
synchronous integration with CSIB, 350-352
Integration Spaghetti, 120
integrity
business exceptions, 119
message integrity, 200-201
process integrity. *See* process integrity
special cases, 119
technical failures versus business exceptions, 118-119
Intelligent Finance, 359
architecture, 362
banking engine services, 371
basic services, 371
implementing SOA, 365
project management, 367-369
repositories, 366
XML, 365-366
lessons learned from implementing SOA, 372-373
project schedule, 364-365
project scope, 360
business impact, 360-361
technology impact, 362-365
service layers, 363-364
technology, 369
architecture, 369-370
contracts, 370
repositories, 370
service interfaces, 370-372
interaction diagram showing check-in process for a Web application, 208
Interface Definition Language (IDL), 20
interface semantics, 42-44
coupling, 48
versus payload semantics, 44-45
document-centric messages, 45-46
interfaces, SOA, 60
intermediary layer, 82
intermediary service, 364
BookAndBill, 93, 98
intermediary services, 76
adapters, 78
façades, 78- 80
functionality-adding services, 81
technology gateways, 76-77
Internet Information Server (IIS), 37
interoperability, off-the-shelf load balancers, 182
Interoperable Object References, 32
isolation, 123
IT, 1
agility, 1
cost savings, 243-244
organizing, 9-10
IT programs versus IT project management, 281
IT project management versus IT programs 281
IT projects versus business projects, 281
IT renovation roadmap, 1
IT steering committee, 368
IT Strategy Mail, 317
Iterative Application Development (IAD), 288

J.D. Edwards, 24
J2EE, 37, 206
software buses, 161
J2ME MIDP 2.0, 223
J2ME SOAP, 224
JAAS (Java Authorization and Authentication Framework), 195
jars, Intelligent Finance, 361
Java Authorization and Authentication Framework (JAAS), 195
JCA (Java Connector Architecture), 185
Just-in-Time production, 24

KPI (key performance indicator), 259

lack of support for long-lived transactions, limitations of 2PC and ACID transactions, 126
languages
COBOL, 16
CORBA, 20
IDL, 20
modeling languages, BPM, 107-108
MODULA, 17
Pascal, 16
SIMULA, 17
layers, 82-83
learning, 10
legacy software, 33
leveraging
SOA to decompose complex systems, 286
thin thread model, 288-289
vertical versus horizontal slicing, 287
SOA to drive development iterations, 289-290
divide and conquer strategies, 290-292
managing parallel iterations, 292-293
limitations
of 2PC
discontinuous networks, 127
integration of legacy systems and packaged applications, 126
lack of support for long-lived transactions, 126
organizational challenges, 126
performance, 125
of ACID transactions
integration of legacy systems and packaged applications, 126
lack of support for long-lived transactions, 126
organizational challenge, 126
performance, 125
load balancers, 181
off-the-shelf load balancers, interoperability 182
local logging, 173
location transparency, B2B, 218
Log Services, 175
log traces, 121
logging, 172
configurations, 177-178
distributed logging, 173-175
execution containers, 169
frameworks, 177-178
local logging, 173
transaction boundaries, 175-177
transaction logs, 176
transaction monitors, 176
logging and tracing, process integrity, 121-122
logic
compensating logic, 153-154
core business logic, 113
process control logic, 113
process logic, 120
logical integration, 343
logic-centric services, 72-73
logs, consolidated logs, 122
loose coupling, 46-49

maintainability, requirements of enterprise software architecture, 6
maintenance, complexity, 242
Manage Evolution, 349
management
CSG, 355
Deutsche Post case study, 322
Winterthur, 338
managing, 9. *See also* organizing IT
parallel iterations, 292-293
service repository, 62
Manhattan Project, 277

KrafzigIXfinal.fm Page 379 Monday, October 18, 2004 11:17 AM

Manifesto for Agile Software Development, 279
Manugistics, 24
Market Units of Winterthur, 326
Martin, James; RAD, 279
Martin, Robert; dX method, 279
MDA (Model Driven Architecture), code generation, 167-168
message integrity, 200-201
Message Queuing systems, 34-35
message transformation, execution containers, 169
Message-Oriented Middleware. *See* MOM
messages, 34
    document-centric messages, 45-46
meta buses, creating, 162
Meta-Object Facility (MOF), 167
Microsoft ASP (Active Server Pages), 37
middleware, authentication, 190-192
middleware heterogeneity, 21-22
minimizing resources for communication on small devices, 224
mitigating risk, motivation for creating SOAs, 249-251
MMS (multimedia message service), 227
MOA, combining with SOA and BPM, 155-156
Model Driven Architecture (MDA), 167-178
modeling languages, BPM, 107-108
MODULA, 17
modularization and component programming, 17
MOF (Meta-Object Facility), 167
MOM (Message-Oriented Middleware), 34-35
    email, 34
    synchronous communication, 39
monitors, TP monitors, 125
motivation, 68
    for creating SOAs, 239-241
        agility, 241-242
        business infrastructure, 246-247
        cost savings, 242-244
        efficient development processes, 247
        evolutionary approach, 248-249
        feedback from, 249
        independence from technology, 245-246
        mitigating risk, 249-251
        reuse, 244-245
MQSeries, 160, 185
MS Visual Basic, VBX components, 17
multi-channel applications, 228-229
    fundamental SOA, 229-230
    process-enabled SOAs, 231-234
    service façades, 230-231
multichannel architecture, 100
multilevel transactions, 127
multimedia message service (MMS), 227
Myers, 16
mySAP, 24

Naming Services, ORB, 20
nested transactions, 127
NetWare Loadable Modules (NLM), 20
networked SOA, 87, 93-98
NLA (Non Life Applications), 329
NLM (NetWare Loadable Modules), 20
Non Life Applications (NLA), 329
Norwegian Computing Center, 17
Novell, NetWare Loadable Modules (NLM), 20
Nygaard, Kristen, 17

Object Management Group (OMG), 167
object orientation, 112
Object Request Broker. *See* ORB
object-oriented programming, 17-18
objects, 17
offsetting, Intelligent Finance, 361
off-the-shelf load balancers, interoperability, 182
OLTP (Online Transaction Processing), 122
OMG (Object Management Group), 167
    CORBA, 160
Online Transaction Processing (OLTP), 122
operating systems, UNIX, 19
operations, update operations, 140-142
    error handling, 143
    sequence numbers, 142
optimistic concurrency control, 135-136
    example, 137-138
Oracle, 24
ORB (Object Request Broker), 20, 33
    Distributed Objects, 32
    Naming Services, 20
organizational challenges, limitations of 2PC and ACID transactions, 126
organizational roadmaps, 10, 281
organizational SOA roadmaps, 261-263
organizing IT, 9-10
orientation, service orientation, 250
out of stock exceptions, 119
outside intervention, 5

parallel iterations, managing, 292-293
Pascal, 16
passenger check-in scenario, 205-207
payload semantics, 43
    coupling, 48
    versus interface semantics, 44-45
        document-centric messages, 45-46
peer-programming, 279
PeopleSoft, 24
performance, limitations of 2PC and ACID transactions, 125
persistent queues, 128
pessimistic concurrency control, 135
    examples, 139
PIMs (Platform Independent Models), 167
Platform Specific Models (PSMs), 167
presentation layer, 327
process and desktop intergration, 344
process control logic, 113
process inconsistencies, 118
process integrity, 118
    2PC, 124-125
    ACID transactions, 122-123
    BPMs, 121, 130-131
    distributed 2PC, 124
    logging and tracing, 121-122
    multilevel transactions, 127
    nested transactions, 127
    persistent queues, 128
    SAGAs, 130
    SOA-driven project management, 293-295
    transactional steps, 128
    transaction chains, 129
    transaction monitors, 124-125
    versus data integrity, 117
    Web service standards, 131-132
process layer, 82
process logic, 120
process management, 104
process orientation, BPM, 104
process-centric services, 74-75, 364
process-enabled SOAs, 87, 98, 100-102
    BPM, 112
        core business logic versus process control logic, 112-114
        design implications, 114
    multi-channel applications, 231-234
processes
    implementing SOA
        at CSG, 347-348
        at Deutsche Post, 316-318
        at Winterthur, 330-332
    in an ideal world, 266-269
Product Lifecycle Management, 24
products, 29
programming paradigms, 18
    component programming, 17
    functional decomposition, 16
    functional programming, 16
    object-oriented programming, 17-18
    service-orientation, 15
project control elements, SOA artifacts, 282-284
project definitions, including service designs, 285-286
project management, 277
    implementing at Intelligent Finance, 367-369
        architecture boards, 368
        DIA, 368
        IT steering committee, 368
        work streams, 368
        XML tsars, 368-369
    implementing SOA
        at CSG, 349
        at Deutsche Post, 319
        at Winterthur, 333-334
project management methodologies
    adding service orientation to, 280
    configuration management, 296
        challenges for, 296-298
        recommendations for the SOA integration, 298-300
    establishing, 278-280

KrafzigIXfinal.fm Page 380 Friday, October 15, 2004 4:32 PM

SOA-driven project management, 281-282
including service designs in, 285-286
leveraging SOA to decompose, 286-289
leveraging SOA to drive, 289-293
process integrity, 293-295
SOA artifacts as project control, 282-284
testing, 301-307
project managers, perspective of SOAs, 253
project schedule, Intelligent Finance, 364-365
project scope
CSG, 342
business impact, 342-343
technology impact, 343-346
Deutsche Post, 312-313
business impact, 313-315
technology impact, 315
Intelligent Finance, 360
business impact, 360-361
technology impact, 362-365
Winterthur, 326
business impact, 326
technology impact, 327-330
projects, success of SOAs, 264-265
PROLOG, 17
protagonists, recommendations for, 274-275
proxy users, 222
PSMs (Platform Specific Models), 167
public enterprise services, 81-82

QoS (Quality of Service), 127
queues, 34
persistent queues, 128

R/2, 23
RAD (Rapid Application Development), 279
RAS (Reliability/Availability/Serviceability), 172
read stability, 125
real world SOAs, 271
example of failure, 271-272
example of success, 272-274
recommendations
for SOA integration team, 298-300
for SOA protagonists, 274-275
reducing risk, 296
Reengineering the Corporation, 103
refactoring enterprise software, architecture, 5
referential integrity, 117
regression test environments, 306
relational databases, 23
Reliability/Availability/Serviceability (RAS), 172
remote procedure call system, SUN-RPC standard, 19
Remote Procedure Calls. *See* RPCs
Remoteness, 19
Rendezvous, 34
repeatable read, 125
repositories
CSG, 354
Deutsche Post case study, 321
implementing
at CSG, 348
at Intelligent Finance, 366
at Winterthur, 332-333
Intelligent Finance, 370
Winterthur, 336-338
requirements of enterprise software architecture, 6-7
resource managers, 124
return on investment (ROI), 259
reusability, requirements of enterprise software architecture, 6
reuse, 242-245
risk, reducing, 296
risk analysis, 295
risk-mitigating effect, motivation for creating SOAs, 249-251
roadmaps
enterprise IT renovation roadmap, 3, 10-11, 262
organizational roadmaps, 10, 261-263, 281
technical roadmaps, 10
ROI (return on investment), 259
RosettaNet, 217
RPCs (Remote Procedure Calls), 31-32
fire-and-forget, 39
RPC-style interfaces, 42
r-tools suite, 19
runtime binding, 63-64
runtime configuration, 178
runtime service discovery based on reflection, 63
runtime service lookup by name, 63
runtime service lookup by properties, 63

SAGAs, 130
SAML (Security Assertion Markup Language), 193
SAP, 23-24
R/2, 23
SBB (Service Backbone), 312, 321
Deutsche Post, 315
scalability, 178-181
CICS, 185
CORBA, 184
EJBs, 182-184
in a heterogeneous SOA, 186
Web Services, 181
wrapped legacy applications, 185-186
SCM (Supply Chain Management), 15, 24
securing SOAs, 187, 200-201
authentication, 187-190
authorization, 193-195
encryption, 196-197
transport security, 196-197
trust domains, 197-198
security
CSG, 355
Deutsche Post case study, 322
execution containers, 168
and heterogeneity, 199-200
J2ME MIDP, 223
lightweight security, 223
Winterthur, 338
Security Assertion Markup Language (SAML), 193
security infrastructures, B2B, 217
security solution, 355
semi-transactional steps, 149-150
separating SOA services, 114
server controlled transactions, 145-146
service access layer, 364
Service Backbone (SBB), 312
service binding, 64
service bus, SOA, 64-65
service buses, 159
service clients, exposing to transaction logic, 145
service contracts, 284-285
service contract iterations, 290
service designs, including in project definitions, 285-286
service dispatchers, 163
service documentation, 250
service enablement, EAI, 212-215
service facades, multi-channel applications, 230-231
service interfaces
CSG, 354
Deutsche Post case study, 322
Intelligent Finance, 370-372
versus services, 371-372
Winterthur, 336-338
service layers
creating layers that replace direct interaction with distributed objects, 214
Intelligent Finance, 363-364
service orientation, 250
adding to project management methodologies, 280
Service Registry, 320
implementing SOA at Deutsche Post, 318-319
service repositories
EAI, 217
managing, 62
SOA, 60-62
service requests, 364
service stubs, 163
code generation, 164
service types, 67
basic services, 70
data-centric services, 70-71
logic-centric services, 72-73
classification, 69-70
intermediary services, 76
adapters, 78
façades, 78-80
functionality-adding services, 81
technology gateways, 76-77
motivation, 68
process-centric services, 74-75
public enterprise services, 81-82
service-orientation, 18
Service-Oriented Architecture. *See* SOA
services, 13-15
basic services
data-centric services, 70-71
logic-centric services, 72-73

KrafzigIXfinal.fm Page 381 Friday, October 15, 2004 4:32 PM

business computing. *See* business computing
cost effectiveness, 14
distributed computing, 19-22
SOAP, 22
XML, 21
domain-specific business services, 327
implementing business processes, 327
intermediary services, 76
adapters, 78
facades, 78-80
functionality-adding services, 81
technology gateways, 76-77
Log Services, 175
Naming Services, ORB, 20
process-centric services, 74-75
versus service interfaces, 371-372
SOA. *See* SOA
technical services, 327
Web Services, 13
World Wide Web, 15
servicing execution containers, 168
session-tokens, 174
short message service (SMS), 81, 227
Siebel, 24
sign-on frameworks, 193
sign-on infrastructures, 190
Simple Object Access Protocol (SOAP), 22
simplicity, requirements of enterprise software architecture, 6
SIMULA, 17
simulated synchronous communication, 39
small devices
designing for, 223-227
minimizing resources for communication, 224
Smith, Howard, 104
SMS (short message service), 81, 227
SMS proxy, 228
SOA (Service-Oriented Architecture), 1, 10, 14, 56-58
application frontends, 58-59
change requests, agility, 89
combining with BPM and MOA, 155-156
EAI, 215
enterprise standards versus architecture, 8
fundamental SOA, 87, 91-92
goals of, 10
heterogeneous SOA, availability and scalability, 186
ideal world
specifics, 269-271
structures and processes, 266-269
layers, 82-83
motivation for creating, 239-241
agility, 241-242
business infrastructure, 246-247
cost savings, 242-244
efficient development processes, 247
evolutionary approach, 248-249
feedback, 249
independence from technology, 245-246
mitigating risk, 249-251
reuse, 244-245
multi-channel applications, 228-229
fundamental SOA, 229-230
process-enabled SOAs, 231-234
service façades, 230-231
networked SOA, 87, 93-98
personal perspective of benefits, 251
architects, 253
CEOs, 252
CIOs, 252
functional departments, 254
project managers, 253
software developers, 254
vendors of standard software, 255
process-enabled SOAs, 87, 98, 100-102
real world, 271
example of failure, 271-272
example of success, 272-274
securing, 187, 200-201
authentication, 187-190
authorization, 193-195
encryption, 196-197
transport security, 196-197
trust domains, 197-198
separating services, 114
service bus, 64-65
service repository, 60-62
services, 59-60
top-down code generation, 166
SOA architects
avoiding distributed 2PC, 143-144
client controlled transactions, 144-145
implicit application level protocols, 146-147
server controlled transactions, 145-146
combining SOA, MOA, and BPM, 155-156
compensating logic, 153-154
example scenario, travel itinerary management, 133-134
optimistic concurrency control
examples, 137-138
implementing, 136
pessimistic concurrency control, examples, 139
transactional steps, 147-149
choosing granularity, 151-153
semi-transactional steps, 149-150
update operations, 140-142
sequence numbers, 142
simplifying error handling, 143
SOA artifacts as project control elements, 282-284
SOA boards, 267-281
SOA infrastructure versus business services, 281
SOA integration team, recommendations for, 298-300
SOA roadmaps, organizational aspects of, 261-263
SOA-driven project management, 281-282
including service designs in the project definition, 285-286
leveraging SOA to decompose complex systems, 286
thin thread model, 288-289
vertical versus, 287
leveraging SOA to drive development iterations, 289-290
divide and conquer, 290-292
managing parallel, 292-293
process integrity, 293-295
SOA artifacts as project control elements, 282-284
SOAP (Simple Object Access Protocol), 22
authentication, 192-193
document-centric messages, 45
J2ME SOAP, 224
software architecture, 55-56
software artifacts, granularity, 284
software assets, 33
software buses, 159-163
CORBA, 160
Enterprise Service Bus, 160
J2EE, 161
software components, 17
software developers, perspective of SOAs, 254
software development, 23
software modules, 17
special cases, 119
specifics of SOAs in an ideal world, 269-271
Spiral Model, 279
Spowart, Jim, 361
SQL, 10
SSB, Deutsche Post, 322
stability, EAI, 215-216
stakeholders, conflicts of interest, 258-261
standards, enterprise software (standards versus architecture), 7-8
Stanford University Network, 19
state comparison, optimistic concurrency controls, 136
stateless semantics, B2B, 218
stateless session beans, 183
static authorization, 194
stress testing, 180
structures
implementing SOA
at CSG, 347-348
at Deutsche Post, 316-318
at Winterthur, 330-332
in an ideal world, 266-269
success
example of SOA failure, 272-274
of SOAs, 263
backers, 265-266
budgets, 264
initial project, 264-265
teams, 265
Sun Microsystems, 19
Enterprise Java Beans, 21
RPCs, 31
SUN-RPC standard, 19
Supply Chain Management (SCM), 15, 24
Sybase, 20
synchronous communication, 38, 350
coupling, 47
MOM, 39
synchronous integration with CSIB, 350-352
synchrony, 38-41

KrafzigIXfinal.fm Page 382 Friday, October 15, 2004 2:08 PM

system management, execution containers, 169
systematic testing, 302

teams, success of SOAs, 265
technical failures, 118
technical integration, 343
technical roadmaps, 10
technical services, 327
technology
    complexity, 241
    CSG, 350
        architecture, 350-355
    Deutsche Post, 320
        architecture, 320-321
        contracts, 322
        management, 322
        repositories, 321
        security, 322
        service interfaces, 322
    Intelligent Finance, 369
        architecture, 369-370
        contracts, 370
        repositories, 370
        service interfaces, 370-372
    Winterthur, 334-336
        contracts, 336-338
        management, 338
        repositories, 336-338
        security, 338
        service interfaces, 336-338
technology gateways, 76-77
technology impact
    CSG, 343-346
    Deutsche Post, 315
    Intelligent Finance, 362-365
    Winterthur, 327-330
technology whitepapers, 267
telnet, 19
test drivers, 306
test suites, creating, 304
testing, 301-307
    functional testing, 301
    regression test environments, creating, 306
    systematic testing, 302
    test suites, creating, 304
thin thread model, 288-289
thin-thread approach, 250
Tibco Software, Rendezvous, 34
tight coupling, 46-49
timestamps, optimistic concurrency control, 136
tokens, 174
tools, automated test tools, 301
top-down code generation, 165-166
Total Quality Management, 103
TP monitors. *See* transaction monitors
TPMs (Transaction Processing Monitors). *See* transaction monitors
tracing, 172
transaction boundaries, logging, 175-177
transaction chains, 129
    combining with compensating transactions, 130
transaction coordinator, 124
transaction logic, exposing to service clients, 145
transaction logs, 176
transaction management, execution containers, 168
transaction monitors, 36, 124-125
    logging, 176
Transaction Processing Monitors (TPMs). *See* transaction monitors
transactional steps, 128-129, 147-149
    choosing granularity, 151-153
    semi-transactional steps, 149-150
transactions, 42
    ACID transactions, 122-123
    client controlled transactions, 144-145
    compensating transactions, 129
    multilevel transactions, 127
    nested transactions, 127
    server controlled transactions, 145-146
    Web services-based transaction protocols, 132
transaction-tokens, 174
transport security, 196-197
trust domains, 197-198
Tuxedo, 21, 36
Two-Phase Commit Protocol. *See* 2PC
TX monitors. *See* transaction monitors

Über bus, 162
UDDI (Universal Description, Discovery and Integration), 46, 49
UN/CEFACT (ebXML), 217
uncommited read, 125
Universal Description, Discovery and Integration (UDDI), 46, 49
UNIX, 19
    workstations, 20
update operations, 140-142
    error handling, 143
    sequence numbers, 142
upgrade ability, EAI, 215-216
user-defined data integrity, 118
users, proxy users, 222

VBX components, 17
vendors of standard software, perspective of SOAs, 255
version counts, optimistic concurrency control, 136
vertical slicing versus horizontal slicing, 287
vision of BPM, 109-111
VT100 systems, 185

Wal-Mart, business computing, 23
Web applications, building, 206, 209-211
Web Service Definition Language (WSDL), 22
Web service standards, process integrity, 131-132
Web Services, 13
    availability, 181
    scalability, 181
whitepapers, 266-267
    business whitepapers, 266
    technology whitepapers, 267
wincoLink, 326
Winterthur, 272, 325-326
    implementing SOA, 330
        processes and structures, 330-332
        project management, 333-334
        repositories, 332-333
    lessons learned from implementing SOA, 339-340
    project scope, 326
        business impact, 326
        technology impact, 327-330
    technology, 334-336
        contracts, 336-338
        management, 338
        repositories, 336-338
        security, 338
        service interfaces, 336-338
Wirth, Niklaus; Pascal, 16
WMS (Workflow Management System), 120
work streams, 368
workflow components, 355
Workflow Management Systems (WMS), 120
workstations, 20
World Wide Web, service, 15
wrapped legacy applications, 185-186
write-test-cases-before-writing-the-actual- code, 279
WSDL (Web Service Definition Language), 22

X/Open DTP (X/Open standard for Distributed Transaction Processing), 125
XA interface, 125
XML, 21
    implementing at Intelligent Finance, 365-366