

## **At Your Service**

**Service-Oriented Computing from an EU Perspective**

**edited by Elisabetta Di Nitto, Anne-Marie Sassen, Paolo Traverso, and Arian Zwegers**

**The MIT Press  
Cambridge, Massachusetts  
London, England**

© 2009 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

For information about special quantity discounts, please email [special\\_sales@mitpress.mit.edu](mailto:special_sales@mitpress.mit.edu)

This book was set in Times New Roman and Syntax on 3B2 by Asco Typesetters, Hong Kong.  
Printed and bound in the United States of America.

#### Library of Congress Cataloging-in-Publication Data

At your service : service-oriented computing from an EU perspective / edited by Elisabetta Di Nitto . . . [et al].  
p. cm. — (Cooperative information systems)

Presents results of collaborative research projects of the European Community's Information Society Technologies Programme about service-oriented computing.

Includes bibliographical references and index.

ISBN 978-0-262-04253-6 (hardcover : alk. paper) 1. Web services—Research—Europe. 2. Internetworking (Telecommunication) 3. Application program interfaces (Computer software) 4. Business enterprises—Computer networks. I. Di Nitto, Elisabetta. II. Information Society Technologies Programme.

TK5105.88813.A85 2009

006.7'6—dc22

2008027498

10 9 8 7 6 5 4 3 2 1

# Index

- Access control, 407–409, 411–412, 415–417, 421, 424–425
- Access policy, 395
- Adaptive behavior, 132
- Alarms, 221
- Ambient intelligence, 10, 358
- Aml. *See* Ambient intelligence
- AMIGO, 44, 50–51, 56, 62
- Annotation, 459, 465, 470–471, 478, 484
- Architecture, 337–339, 392
  - and behavioral constraints, 247
  - and detailed design model, 247
  - model, 246–247
- Architecture-based service discovery, 242, 245
- Architecture Time Discovery, 260
- Aspect-oriented Software Development (AOSD), 136
- ASTRO, 183–185, 191, 196, 204–205, 207–210
- ATHENA
  - CBP framework, 459
  - framework, 457, 464
  - reference architecture, 458–459, 478
- Auction type, 254
- Audit, 395
- Authentication, 392, 408, 411–412, 414–415
- Authorization, 408, 411–421, 424–425
- Automated SLA negotiation, 254. *See also* Service level agreement
- Autonomic computing, 16, 131. *See also* Self, adaptive software
- Availability, 8, 393, 402, 489–490, 498–500, 506–507, 512
- Axis SOAP engine, 493, 497, 506, 509
  
- Banca Tesoriere, 185, 187, 188, 191, 192–193, 195, 196, 199, 206, 207
- Base service, 279, 283–285, 287, 289–290, 292–297.  
*See also* Basic service
- Basic service, 276–278. *See also* Base service
- Behavioral models of services, 247
- Binding, 253
  - constraints, 243
  - global binding, 257
  - global rebinding, 254
  - (re)binding strategy, 258
- Biologically inspired approaches, 9, 38–40. *See also*
  - Biologically inspired computing
  - Biologically inspired computing, 16. *See also* Biologically inspired approaches
- BPEL (Business Process Execution Language), 243, 260, 276, 284, 289, 363, 379, 381. *See also*
  - Orchestration; WS-BPEL
  - ActiveBPEL, 183, 188, 191, 203, 206–207
  - augmented BPEL process, 244
  - mapping to SRML, 170
  - workflow, 253
- Broker, 330, 336, 342, 410, 422–423, 425
- BSI ISO/IEC 20000, 387–388
- Business logic, 184, 187
- Business model, 275, 283, 296
- Business process, 159, 161
  - engine, 461, 467
  - management, 394, 524–527
  - model, 466, 469–471, 487
  - modeling language, 459
  - optimization, 216
  
- CA. *See* Context awareness
- Calculus for the Orchestration of Web Services, 160, 174–177
- Capability, 329, 332, 349–350
  - provided, 499–502, 504, 511–512
  - requested, 499–501, 503–505, 511–512
- Case-based memory. *See* Organizational memory
- CER. *See* Context enabler and reasoner
- Choreography, 331, 335
- Collaboration, 104, 123, 125. *See also* Unified modeling language, collaboration
- Collaborative behavior, 9
- Compatibility, 115–116
  - guarantees, 105. *See also* Service validation
- Compensation, 166
- Compliance, 388
- Component service, 183–184, 191, 195, 199, 202

- Composite service, 183–184, 191, 195, 196–198, 199, 203, 209, 242, 256, 276, 282, 285. *See also* Service composition
- Composition. *See* Service composition
- Consistency, 115–116. *See also* Service validation
- ConSpec language, 433
- Constraint semirings, 173
- Content-based subscription, 257
- Context, 358, 380–382
  - enabler and reasoner, 293
  - in service composition, 378–379
  - specification of, 362, 366, 368–369, 372
  - tool-aided modeling of, 374–375
  - and usage feeds, 293–294
- Context awareness, 131, 281–282, 284–285, 292–293, 296
- Context-related conditions, 252
- Continuous-time Markov chain, 177
- Contract management, 6
- Contract-policy matching, 439
- Control requirements. *See* Requirements, control
- COWS. *See* Calculus for the Orchestration of Web Services
- Credential, 412, 414, 421, 424
- Crisis management, 67–69, 82, 97
- Cross-cutting concern, 490, 492
- Cross-organizational business process, 457–458, 465–466, 468–469, 475
- Cryptography, 392
- C-semirings. *See* Constraint semirings
- CTMC. *See* Continuous-time Markov chain
- CUF. *See* Context and usage feeds
- Cybercrime, 399
  
- DAML-S, 264
- Data analyzer, 258
- Data collector, 258
- Data net, 196–198
- Data requirements. *See* Requirements, data
- DeDiSys, 490, 498, 500, 512
- Delivery manager, 256
- Dependability, 10, 387–389, 400, 489–491, 498, 505, 510, 512
- Dependable building blocks, 399
- Device identity, 391
- Diagnosis, 221, 223
  - consistency-based, 229
  - decentralized, 229
  - diagnostic algorithm, 228
  - global, 224
  - local diagnoser, 223
  - model-based (MBD), 228
- DIP, 327, 330, 342, 353
- DIRE. *See* SeCSE, distributed registry
- Discovery. *See* Service discovery
- Distributed communication bus, 256
- Distributed business process, 183–184, 205. *See also* Cross-organizational business process
- Distributed computing, 407
- Distributed registries, distributed
  - Distributed SWS repository (DSWS-R), 302, 316–317. *See also* Semantic Web Service, persistence and publication
- Due diligence, 388
- Dynamic reconfiguration, 214
  
- Early service discovery (ESD). *See* Service discovery, early
- e-Banking, 183
- ebXML, 71, 81, 84, 260–261, 266
- Eclipse, 80
- e-Commerce, 183, 210
- Education, 398
- e-Government, 183–185, 187, 207, 210, 327, 330, 341
- Emergency
  - officer, 328, 342, 345
  - management system, 341–342
  - situation, 328, 341, 347
- End-to-end quality provision, 525
- Enterprise modeling, 458–459, 461, 463, 465, 481, 487–488
- Enterprise Service Bus, 260
- e-Procurement, 459–460, 468, 484–485
- ESB. *See* Enterprise service bus
- Essex County Council (ECC), 327, 342, 345, 350
- Event-condition-action (ECA), 253
- Executable script, 279, 289, 292. *See also* BPEL; WS-BPEL
- Extrafunctional behavior, 134
  
- Facets, 3, 244–245, 255–257, 260–263, 282–285, 290, 292
- Failure, 213
- Faults, 220
  - catalogue, 221
  - functional, 220
  - identification, 231
  - permanent, 232
  - QoS, 220
  - temporary, 232
- Federate registries. *See* Registries, federate
- Federation, 257
- Flow requirements. *See* Requirements, flow
- Formal analysis, 160–161, 163, 177, 179
- Functional requirements. *See* Requirements, functional
- Future Internet, 519, 520
  
- Generic service model, 72, 74–77, 81, 97
- Genetic algorithms, 255
- Global binding. *See* Binding, global
- Global computer, 159
- Global rebinding. *See* Binding, global rebinding
- Globus, 412–415, 425, 505, 510
- Goal, 104, 113–114, 329, 331–337, 339–343, 344, 346–350

- Governance, 395, 398
- Graphical composition. *See* Service composition, graphical
- Graphical service description. *See* Service description, graphical
- Grid computing, 160
- Grid services, 4, 9, 70–71, 74, 77, 79–80, 85, 87, 89, 97  
WSRF, 70, 89
- Grounding, 329, 335
- Health care, 67, 69, 97
- Heterogeneous registries. *See* Registries, heterogeneous
- Heterogeneous services, 67–74, 78, 80, 84, 81–89, 94, 96–97
  - composition (*see* Service composition, heterogeneous)
- Hosting environment, 410–411, 425
- Hybrid systems, 241
- Identification, 392
- Identity management, 403
- INDISS, 50–53, 56–58
- Industrial adoption, 10
- Information filtering, 23
- Information security, 393
- Information Society Technologies (IST), 2
- Infrastructure layer, 5, 9, 15
- INFRAWEBS framework, 299–325
  - advantages of, 320–321
  - assumptions and shortcomings of, 321
  - conceptual architecture of, 301–303
  - evaluation of, 318–321, 323
  - implementation of (*see* Integrated INFRAWEBS framework)
  - pilot application of, 318–320
- Integrated INFRAWEBS framework (IIF), 317–318
- Interface, 104, 109, 113–114
- Interoperability, 4, 7, 9, 10, 357–360, 380–382
  - on application layer, 362–363, 376
  - artifact, 458
  - framework, 457, 487
  - issue, 457, 458
  - on middleware layer, 362–364, 376
  - semantic interoperability, 465, 476
- Invocation, 335. *See also* Choreography; Orchestration
- IRS-III, 330–340, 344, 347, 350, 352
- ISO 17799, 394
- ISO/IEC 27002, 387, 388. *See also* ISO 17799
- IT governance, 393
- IT infrastructure, 159
- ITIL, 394
- J2EE, 188, 292, 294–295
- JAIN-SLEE, 292, 294. *See also* Mobicents
- Java ME, 443
- Java Virtual Machine (JVM), 146
- Jena, 372, 374
- Jini, 257
- JSLEE. *See* JAIN-SLEE
- LDAP, 71
- Legacy systems, 330, 337, 344, 350
- Legal requirements. *See* Requirements, legal
- Long-running transactions, 166
- Loose monitoring. *See* Monitoring, loose
- MADAM, 129
- Maintainability, 402
- MAIS, 264
- Management services, 6, 490
- Mashups, 274, 277–278
- Matching (of service properties), 500, 501, 505–506, 510–512
- Mathematical foundations, 9, 159–161, 177, 179
- Mediation, 328–329, 331–337, 344–352
- Message relaying, 21–23
- METEOR-S, 264
- Methodology, 328, 339
- Middleware, 129
  - for the smart home, 357–358, 363–365, 367, 377 (*see also* Interoperability, on middleware-layer)
- MIDP, 443
- Mixed initiative, 110
- Mobicents, 292. *See also* JAIN-SLEE
- Mobile application life cycle, 432
- Mobile computing, 131
- Mobile service, 104
- Model
  - platform-independent (PIM), 146
  - platform-specific (PSM), 146
  - transformation, 161
- Model-based diagnosis. *See* Diagnosis, model-based
- Model-driven architecture (MDA), 121, 463, 466
- Model-driven development (MDD), 146
- Model-driven interoperability, 458, 465
- Model-driven security, 396–397
- Model-driven service engineering, 101, 121
- Modeling, 109, 112–114, 121
  - behavior of SOA, 166
  - at business level, 457, 460, 462, 466, 469–471, 473, 480–482
  - at execution level, 462, 467, 470–473, 482, 484
  - formal modeling, 109
  - structure of SOA, 164
  - at technical level, 459, 466, 470–474, 481–485
- Monitoring, 183–185, 205–210, 226, 445
  - component, 253
  - of functional and QoS properties, 258
  - of services, 241
  - loose, 260
  - manager, 258
  - mechanisms, 258
  - platform, 260
  - rule, 258–259
  - tight, 260
- MUSDAC, 50, 54–58, 58–63

- Natural language, 393
  - requirements (*see* Requirements, natural language)
- Negotiation (of service properties), 511
  - reference implementation, 521
- NESSI, 2, 4, 241, 391, 517, 521
  - adoption, 517, 520
  - framework, 517, 519
  - holistic model, 517–520
  - landscape, 517, 520
  - taxonomic framework of, 4
- NESSI Open Service Framework, 517, 520–522
  - reference architecture, 521
- .NET, 294
- Network architecture, 20–22
- NEXOF. *See* NESSI Open Service Framework
- Next Generations Grids, 407
- Nomadic users, 10
- Nonfunctional requirements. *See* Requirements, nonfunctional
  
- OCML, 331–334, 337, 339–341, 345–347, 350
- Ontology, 17, 36–37, 105, 327, 329–331, 358–382, 399.
  - See also* Semantic service
  - archetype, 346–348
  - context, 346–348
  - domain, 335, 346–348, 352
  - GUI, 346–348
  - IRS-III Service Ontology, 333 (*see also* IRS-III)
  - spatial, 346–348
- Open Grid Services Architecture, 408
- Opportunistic networking, 16, 21–22
- OPUCE, 273–298
- Orchestration, 223, 276, 290, 294, 296, 329, 335–336, 348, 352. *See also* BPEL; Service orchestration
- Organizational memory (OM), 302, 314–316
- OWL, 358, 361–362, 372–376
- OWL-S, 244, 263, 328, 331, 362–368
  
- P2P Service Description Language (PSDL), 77
- P2P Services, 67–72, 77, 79, 83, 85, 89, 95, 96
  - JXTA, 71, 77, 81, 83–84
- Patterns, 390, 394, 397
- PAWS. *See* Processes with Adaptive Web Services
- PEPA. *See* Performance Evaluation Process Algebra
- Performance Evaluation Process Algebra, 160, 177
- Pervasive computing environments, 13, 15, 43, 46, 50, 53–54, 58, 62
  - services, 43, 429
- PLASTIC, 44, 53, 56, 62
- Platform-independent model. *See* Model, platform-independent
- Platform-specific model. *See* Model, platform-specific
- Policy, 392
  - decision point, 421, 425, 445
  - information service, 446
  - languages, 399
  - refinement, 422
- Portal, 275, 279, 281, 285–287, 292–293, 296
  
- Privacy, 398, 402, 408, 415
- Proactive computing, 131
- Process
  - abstraction, 459–463
  - enactment, 475, 479
  - execution, 459, 461, 484, 487
  - private, 459–463, 467–468, 470–471, 473, 476, 478, 482–484, 487
  - view, 462–463, 465, 467–468, 470–471, 476, 478, 482–487
- Process calculus, 174
- Process-oriented service, 183–184
- Processes with Adaptive Web Services (PAWS), 216, 236
- Property prediction, 139, 144
- Prosumer, 273
- PSDL. *See* P2P Service Description Language
- Publish and subscribe middleware, 256
- PYRAMID-S, 266
  
- QoS. *See* Quality of service
- Qualitative analysis. *See* Formal analysis
- Quality of service (QoS), 6, 8, 68, 71, 74, 95, 129, 207, 209–210, 214, 241, 255, 358, 381–382, 490, 498–499, 511, 528
  - adaptive, 216
  - attributes, 244, 257
  - constraints, 247, 257–258
  - contract, 225
  - faults, 220, 238
  - Globus Architecture for Reservation and Allocation (GARA), 71
  - level, 255
  - metrics, 6
  - monitor, 303, 306 (*see also* Semantic Web Service, monitoring of)
  - ontologies, 244
  - parameters, 236
  - optimization and reconfiguration, 216
  - properties, 247
  - requirements, 214
  - in service composition, 214, 378–379
  - single service-declared QoS, 258
  - specification of, 362, 366–372
  - value, 258
  - violation, 6
  - WSLA, 71
  - WSOL, 71
  - WS-QoS, 71, 84
- Quality of service tool (QST), 259
- Quality requirements. *See* Requirements, quality
- Quality properties, 247
- Quantitative analysis. *See* Formal analysis
  
- Recovery, 221
  - action, 254
  - global, 223
  - local, 223

- Redundancy, 498
- Registries, 256, 260
  - distributed, 257
  - federate, 260
  - heterogeneous, 256
- Regression testing tool, 255
- Reliability, 8, 400, 402, 489, 495
- Remote service management, 293–294
- Repair actions, 227
  - compensate, 232
  - flow-level, 227
  - infrastructural, 227
  - redo, 232
  - retry, 232
  - service-level, 227
  - substitute, 232
- Repair plans, 231
  - execution, 232
- Replication
  - active, 491, 494, 509
  - coordinator-cohort, 492
  - manager, 492–493, 495–496, 509
  - middleware, 491–493, 495–497, 509, 512
  - primary-backup, 491–497, 509
  - protocol, 489, 493, 495–496, 509
- Repository. *See* Service repository
- Reputation, 411–412, 416, 418, 421–425
- Requirements, 246, 408, 414, 418, 422, 424–425
  - control, 184, 193, 195, 199, 203
  - data, 184, 193, 195–196, 196–199
  - and discovery activity, 245
  - flow, 184, 193, 198–199, 203
  - functional, 242, 244, 252
  - legal, 387
  - natural language, 246
  - nonfunctional, 242, 244, 252
  - quality, 252
  - security, 10, 387–389, 391, 396–397
  - and service discovery, 245
- Resilience, 398, 401
  - engineering, 400–401
- Resource broker, 422–423, 425. *See also* Broker
- Return on Security Investment (ROSI), 394
- RFID (radio frequency identification), 17, 18
- Risk assessment, 397, 401
- Risk impact, 394
- RSM. *See* Remote service management
- RTML. *See* Runtime Monitoring Specification Language
- Rule set, 258
- Runtime enforcement, 443
- Runtime environment, 260
- Runtime Monitoring Specification Language (RTML), 206–208, 210, 267
- Runtime service discovery, 245, 252
- SA. *See* Service advertising
- Safety, 402
- SCA. *See* Service component architecture
- SCS. *See* Service-centric system
- S-cube, 523
  - architecture, 526
- SDE. *See* SeCSE, development environment
- SECMOL. *See* Service-centric Monitoring Language
- SeCSE, 241, 327, 447
  - approach, 242
  - conceptual model, 2–3, 242
  - development environment, 259, 260
  - development process, 242
  - distributed registry, 256
  - facets (*see* Facets)
  - integrated platform, 242
  - methodology, 242
  - monitoring subsystem, 258
  - negotiation platform, 254
  - process, 244
  - registry, 244, 260
  - service registry (*see* SeCSE, registry)
- Security, 387, 389
  - attributes, 8
  - built-in, 387
  - by contract, 10, 431, 444
  - contract, 431, 433
  - end-to-end, 404
  - engineering, 395–397
  - functions, 392
  - governance, 395
  - management, 393
  - mechanism, 388–389
  - patterns, 390–391, 397
  - policy, 392–393, 432
  - properties, 391, 394, 396
  - requirements (*see* Requirements, security)
  - sensitive, 395, 397
  - services, 389, 392
  - solutions, 390–391
  - and privacy, 303
  - and trust, 6
- SEE. *See* Service execution, environment
- Selection, 332, 347
- Self
  - adaptive service compositions, 241
  - adaptive software, 136, 144, 148
  - configuration policy, 253
  - healing, 10
  - healing Web Service, 213
  - management (*see* Service autonomy)
  - SH-BPEL, 236
- Semantic
  - annotation, 184 (*see also* Annotation)
  - descriptions, 332, 335, 348 (*see also* Semantic Web Service, descriptions)
- Semantic information router (SIR), 302–304
- Semantic layer, 5, 328. *See also* Semantic Web Service, layer
- Semantic Web, 327, 328

- Semantic Web Service (SWS), 299–325, 327–330, 341, 343, 345, 353. *See also* Service application consumer, 300–301, 306 application provider, 300–301, 305–306 brokers or aggregators, 300–301, 304–305 composer (SWS Composer), 302, 304–305 composition (design-time) of, 304–305 composition (runtime) of, 306–307 description language, 362–372 descriptions, 344, 347, 352 description vocabulary, 361–362 designer (SWS Designer), 302, 304, 307–317 design of, 303–304, 307–317 developer, 331, 339 discovery of, 306 (*see also* Service access middleware) execution of, 306–307 executor (SWS Executor), 303, 306–307 layer, 338 life-cycle of, 300–301 modeling, 358–359 monitoring of, 306 (*see also* Quality of service, monitor) persistence and publication of, 316–317 (*see also* distributed SWS repository) specification, 263 visual modeling tool, 372–376
- Semantics, 6, 8, 10, 116, 121–122, 276–277, 283–285, 292, 297, 399, 457–459, 464–465, 470, 476
- Sensoria, 159, 161 approach, 161 automotive case study, 163–164 development environment (SDE), 161–162 reference modeling language (SRML), 160, 167–172 research themes, 162
- Separation of concerns, 9, 129
- Service functional properties of, 363–366 nonfunctional properties of, 366–372
- Service access middleware (SAM), 302, 306, 323 (*see also* Semantic Web Service, discovery of)
- Service adaptation, 26, 278, 293–294, 296–297, 376–379, 524, 526 methodologies, 525 and monitoring, 528
- Service advertising, 278–279, 281–282, 286, 292–294, 296–297
- Service analysis, 163, 176
- Service announcement, 46–48, 55
- Service architecture, 102, 104, 389, 401 client-server, 104 peer-to-peer, 104
- Service autonomy, 29–31
- Service availability, 391
- Service behavior, 102, 115, 123–124
- Service bus, 394
- Service candidate, 246
- Service capabilities, 363
- Service Centric MONitoring Language (SECMOL), 258–259
- Service-centric system (SCS). *See* Service-oriented system
- Service component architecture, 160, 167
- Service composition, 8–9, 68, 72–73, 77–85, 94–96, 102, 104–105, 120, 183–185, 191–193, 199, 203–205, 209–210, 253, 277, 279, 287, 293, 295, 328, 347, 376–379, 394, 524, 526. *See also* Composite service; Orchestration abstract service composition, 91 BPEL4WS, 72, 75 (*see also* WS-BPEL) composition suite, 73, 77, 79–81, 88, 91, 94 concrete service composition, 94 and coordination, 527 design tool, 260 graphical composition, 279, 289, 292 heterogeneous service composition, 73, 80, 85, 96 language, 253 layer, 10 methodology, 89–90, 94 requirement, 183–184, 201, 209–210 visual service composition, 77, 94. Visual Service Composition Language (VSCL), 72–73, 77–81, 86, 94–95, 97
- Service consumer, 243–244, 255, 274, 277, 499–500, 506, 511
- Service conversations, 363
- Service creation, 273–282, 286–294, 296, 298
- Service delivery, 256
- Service deployment, 163, 273–274, 276–278, 281, 283, 285, 289–294
- Service description, 3, 9–10, 45–50, 55, 57–60, 62, 282, 287, 290–292, 294, 331–337, 339. *See also* Web Service modeling ontology, descriptions; Semantic Web Service, descriptions graphical service description, 289 life cycle, 242
- Service directory, 260
- Service discovery, 4, 9, 68, 71–72, 83–84, 92, 95–96, 116–119, 120, 123, 277, 294, 328, 332, 347, 376–379 early service discovery (ESD), 259 or dynamic binding, 241 process, 247 protocols, 43, 45–47, 50–61, 63 query, 247 request, 46, 50, 55–56, 58–59, 61 service, 501, 504, 510, 512
- Service disruption, 394
- Service economy, 517, 520
- Service emergence, 33–34
- Service engineering, 8, 10, 525, 527, 242, 253
- Service evolution, 38–39
- Service execution, 273–277, 281, 285, 293–294, 376–379 environment, 275, 279, 282, 292–297
- Service infrastructure, 524, 527 layer, 9
- Service index, 504–506, 510–512
- Service initiation, 35–37
- Service instance, 410–411



- Service integration, 5, 37–38, 255
- Service layer, 337, 345
- Service level agreement, 3, 145, 162, 172, 177–179, 242–243, 255, 395, 399
  - compliance, 525
  - template, 256
  - testing, 256
- Service lifecycle, 274–275, 278–280, 282, 286, 292–294, 296
  - management, 6, 279, 282, 285–286, 289–292, 294, 296–297
- Service logic, 275–276, 282, 284–285, 287, 291–292, 294–296
  - engine, 295–296
- Service management, 281, 292, 294, 528
- Service matching, 376–379
- Service metrics, 528
- Service model, 75, 77. *See also* Generic service model
  - points of view, 74–75, 77
- Service modeling, 8, 102, 124, 162, 164–166. *See also* Modeling
- Service monitoring, 526
- Service offer, 254
- Service orchestration, 166, 174–175, 523
- Service personalization, 278, 281, 293, 297
- Service provisioning, 43, 45–46, 48–50, 253, 255, 273, 274, 277–280, 282–283, 285, 289–292, 330, 331, 342, 409, 410–411, 423, 425, 498–507, 511
- Service Regression Testing tool (RTT), 259
- Service repository, 43, 45–46, 48–49, 279, 287, 290–292
- Service request, 43, 254, 331, 410–411, 423
- Service retirement, 39
- Service sharing, 278, 280, 283, 285–286, 292, 294
  - service specification, 3, 241, 244
  - tool (SST), 260
- Service subscription, 281, 290, 292
- Service user, 280, 285
- Service validation, 115–116
- Service orientation, 241
- Service-oriented application. *See* Service-oriented system
- Service-oriented architecture, 17, 102, 119–122, 255, 260, 276–278, 296–297, 328, 339, 447, 523. *See also* Service-oriented system
  - limitation of, 102
- Service-oriented computing, 1, 67, 72, 96–97, 159, 241, 255
- Service-oriented development. *See* Service-oriented computing
- Service-oriented infrastructure, 401
- Service-oriented system, 1, 183, 205, 241–242, 246, 252–253, 257, 260, 520, 525
  - data-centric, 490–491, 498, 512
  - development, 245
  - engineering process, 244
  - issues and challenges, 1
  - resource-centric, 490, 498, 512
- SerWorks, 24–25
- SH-BPEL. *See* Self healing, SH-BPEL
- Simple Service Discovery Protocol (SSDP), 46–48, 50, 53–54
- SLA. *See* Service level agreement
- SLE. *See* Service logic, engine
- SLM. *See* Service lifecycle, management
- Smart home environment, 357–358, 379–382
- SOA. *See* Service-oriented architecture
- SOC. *See* Service-oriented computing
- Social network, 280, 286
- Sociology, 398
- SODIUM, 263
- Soft Constraints, 173–174
- Software adaptation, 133. *See also* Adaptive behavior
- Software Development Process usability for various, 161
- Software infrastructure, 520–521
- SPARQL, 374
- Special-purpose subscription, 257
- Specification and Description Language (SDL), 109, 116
- SRML. *See* Sensoria, reference modeling language
- SSDP. *See* Simple Service Discovery Protocol
- Stakeholders, 327, 339, 350–351, 353
- Standard facets. *See* Facets
- State machine, 113
- State transition system (STS), 188–190, 194, 196, 201, 207
- State, 109
- Stateful service sessions, 108
- Static analysis, 432, 435
- Static and dynamic discovery, 245. *See also* Service discovery
- STREAM Flows! System. *See* INFRAWEBBS
  - Framework, pilot application of
  - STS. *See* State transition system
- SWS. *See* Semantic Web Service
- Syntactical matching methods, 263
- System Information Service, 446
- System integrator, 242–243, 254
- TARSU (TAssa sui Rifiuti Solidi Urbani, municipal solid waste tax), 185–187, 188, 191, 195, 198–200, 203, 208
- Testing, 241
  - facets (*see* Facets)
  - tools, 259
- Third parties, 273, 275, 278–279, 282, 297
- Threat, 398
- Tight monitoring. *See* Monitoring, tight monitoring
- Time management, 226
- Tool-aided semantic service modeling. *See* Semantic Web Service, visual modeling tool
- Topic-based subscription, 257
- Trading, 500–502, 506–507
- Trust, 335, 353, 397
  - management, 399, 408, 412, 424
  - mechanism, 403
  - model, 399
  - trust infrastructures, 392
- Trustworthiness, 398

- Ubiquitous computing, 131, 134
- UDDI. *See* Universal Description, Discovery and Integration
- Ufficio Ragioneria, 185, 187, 188, 191–195, 196–199, 207–208
- Ufficio Tributi, 185, 188–190, 195, 196–199, 207–208
- UIM. *See* User information management
- UML. *See* Unified Modeling Language
- UMLSec. *See* Unified Modeling Language, UMLSec
- Unified Modeling Language, 77–78, 95, 111–112, 123, 146, 160, 161–163, 167, 247, 396
  - collaboration, 102, 104, 112–113
  - profile, 78, 116, 125
  - semantics, 396
  - service-oriented extensions for, 164–167
  - stereotypes for SOA, 164–167
  - UMLSec 396
- Unified Service Composition, 72, 85. *See also* Service composition, heterogeneous
  - ECA rules, 86
  - USCL, 72–7, 77, 80, 85–88, 94–95
  - USCL engine, 88–89, 96
- Unified Service Discovery, 81. *See also* Service discovery
  - Unified Service Query Language (USQL), 72–73, 77, 80–85, 88, 92–93, 95, 97
  - USQL engine, 81, 83–85, 88, 92–95, 97
- Universal Description, Discovery and Integration (UDDI), 46–47, 49–50, 71, 81, 84, 244, 260–262
- UPnP services, 68
  - sensor services, 68
- Usability, 398
- Usage control, 407–408, 411, 416, 418, 420–422, 424–425
- User, 330, 342
  - centricity, 273–274, 277–280, 285, 296–297, 403
  - confidence, 402
  - context, 274, 280–281, 293–294
  - information management, 279, 281, 286, 289, 291, 293, 296–297
  - profile, 273, 281, 286, 293–294
  - sphere, 280, 285
- USQL. *See* Unified Service Discovery, Unified Service Query Language
- USQL engine. *See* Unified Service Discovery, USQL engine
- Utility, 129
  - functions, 144, 153
- Validation, 241, 255
- Variability model, 141
- Variation point, 144
  - management, 253
  - realization, 253
- Verification, 255
- Virtual meeting place, 106
- Virtual organization, 407–412, 414, 425
- Visual service composition. *See* Service composition, visual
- VSCL. *See* Service Composition, Visual Service Composition Language
- VSCL2USCL translator, 80, 88, 94
- Weakness, 389, 404
- Web 2.0, 273–274, 277–278, 294, 296–297
- Web-based system, 255
- Web Ontology Language. *See* OWL
- Web service level agreement, 267
- Web Service Modeling Language (WSML), 299, 309–312
- Web Service Modeling Ontology (WSMO), 299–300, 307, 322–323, 328–329, 331, 335, 339
  - descriptions, 307, 344, 347 (*see also* Semantic Web Service descriptions; Service description)
  - orchestration, 307 (*see also* Orchestration)
  - service capability, 307, 309–312
- service choreography, 307, 309–312 (*see also* Choreography) studio, 339–340
  - top-level concepts, 307
- Web Service, 4, 9, 67–94, 160, 241, 243, 263, 278, 290, 294, 327–329, 331, 335, 341–348, 352–353, 412. *See also* Service
  - complex Web Services, 213
  - process calculus for the orchestration of, 174
  - providers, 300–301, 303–304
  - selection, 263
  - SOAP, 71
- Web Service Description Language (WSDL), 70–71, 77, 81, 84, 94, 183, 188, 191, 203, 242, 244, 262–263
  - WS-Agreement, 258
  - WS-Discovery, 48, 56
  - WS-Policy, 263, 500
- Well-formedness, 115–116. *See also* Service validation
- Wireless sensor networks, 18, 20
- WordNet, 246, 263
- Workflow, 252, 409–410
- WS-BPEL (Business Process Execution Language), 95, 183–184, 189, 191, 195, 201, 204, 205, 208, 225, 253. *See also* BPEL
  - abstract WS-BPEL, 183, 189, 191, 195, 201
  - executable WS-BPEL, 183–184, 191
- WSDL. *See* Web service; Web Service Description Language
- WSLA. *See* Web service level agreement
- WSMO. *See* Web Service Modeling Ontology
- XML, 183, 188, 199, 209, 259–261, 282–283
- Xpath, 261
- Xquery, 261
- XSL, 204, 210n4
- XSLT, 204, 210n4, 260