Appendix A: The ATM System

The ATM system was partly specified in Chapters 2 and 3. The entire system, to an appropriate level of detail, is shown below.

Figure A.1. The system map of the ATM system
Consortium refers to Figure A.2. SO, the system diagram of the ATM system

SD Paragraph:

Consortium consists of 5 Banks.
Bank holds at least one Account.
Customer owns Account.
Customer owns 1 to 2 Cash Cards.
Cash Card accesses Account.
Transaction refers to Account.
Customer handles Transaction Executing.
Transaction Executing requires ATM, Cash Card, Consortium, and Bank.
Transaction Executing affects Account and Cash.
Transaction Executing yields either Transaction or Denial Notice.
Transaction Executing from SD zooms in SD1 into Account Checking, Transaction Processing, and Notifying, as well as Approval and Card Data.

Figure A.3. SD1 of the ATM system

SD1 Paragraph:
Customer handles Transaction Executing.
Transaction Executing requires ATM.
Account Checking yields Approval and Card Data.
Account Checking requires Bank, Cash Card and Consortium.
Approval can be obtained or denied.
Transaction Processing occurs if Approval is obtained.
Transaction Processing requires Bank and Card Data.
Transaction Processing affects Account and Cash.
Transaction Processing yields either Transaction or Denial Notice.
Notifying occurs if Approval is denied.
Either Notifying or Transaction Processing yields Denial Notice.
Account Checking from SD1 zooms in SD1.1 into Cash Card Validating, Password Checking, and Approval Denying, as well as "Cash Card is Valid?"

SD1.1 Paragraph:

Account Checking requires ATM.
Customer handles Cash Card Validating and Password Checking.
Cash Card Validating requires Consortium, Bank, and Cash Card.
Cash Card Validating yields Card Data.
Cash Card Validating determines whether Cash Card is valid.
Cash Card Validating and Password Checking require Consortium.
Password Checking occurs if Cash Card is valid, otherwise Approval Denying occurs.
Password Checking requires Consortium, and Card Data.
Password Checking yields Approval.
Approval can be obtained or denied.
Approval Denying yields denied Approval.
Cash Card from SD1.1 unfolds in SD1.1.1 to exhibit Bank Code and Account Number.

Cash Card Validating from SD1.1 zooms in SD1.1.1 into Card Reading, Bank Code Checking, Account Number Checking, Access Denying, and Cash Card Approving, as well as “Bank Code is Valid?” and “Account Number is Valid?”.

Figure A.5. SD1.1.1 of the ATM system

SD1.1.1 Paragraph:
Bank Code identifies Bank.
Cash Card Validating requires ATM.
Customer handles Card Reading.
Card Reading requires Cash Card.
Card Reading yields Card Data.
Bank Code Checking determines whether Bank Code is valid.
Bank Code Checking and Account Number Checking require Card Data.
Account Number Checking occurs if Bank Code is valid, otherwise Access Denying occurs.
Account Number Checking requires Bank, Account Number, and Card Data.
Account Number Checking determines whether Account Number is valid.
**Cash Card Approving** occurs if **Account Number is valid**, otherwise **Access Denying** occurs.

**Cash Card Approving** determines that **Cash Card is valid**.

**Access Denying** occurs either if **Bank Code is not valid** or if **Account Number is not valid**.

**Access Denying** determines that **Cash Card is not valid**.

**Account** from SD1.1 unfolds in SD1.1.2 to relate to **Password**.

**Password Checking** from SD1.1 zooms in SD1.1.2 into **Number Of Trials Initializing**, **Password Requesting**, **Password Keying**, **Password Comparing**, **Trial Comparing**, **Approving**, **Confiscating**, and **Incrementing**, as well as **Number Of Trials**, **Max Trials**, **Loop**, **Keyed in Password**, "**Number Of Trials is greater than Max Trials?**", and "**Keyed in Password is correct?**".

---

**Figure A.6. SD1.1.2 of the ATM system**
Password Requesting from SD1.1.2 zooms in SD1.1.2.1 into Message Displaying and Reading, as well as Displayed Message.

ATM from SD1.1.2 unfolds in SD1.1.2.1 to consist of Keyboard, CPU, Screen, and additional parts.

Customer handles Password Keying and Reading.
Password Requesting requires Screen.
Loop can be uninitialized or initialized.
Message Displaying occurs if Loop is initialized, and if Number Of Trials is not greater than Max Trials.
Message Displaying requires CPU.
Message Displaying yields Displayed Message.
Reading requires Screen.
Reading invokes Password Keying.

Figure A.7. SD1.1.2.1 of the ATM system
Transaction from SD1 unfolds in SD1.2 to exhibit Amount and Type. Cash from SD1 unfolds in SD1.2 to exhibit Ownership.

Transaction Processing from SD1 zooms in SD1.2 into Transaction Querying, Credit Limit Checking, Cash Dispensing, Cash Accepting, and Denial Notifying, as well as "Credit Limit is Exceeded?"

Figure A.8. SD1.2 of the ATM system

SD1.2 Paragraph:
Type of Transaction can be withdrawal or deposit.
Cash exhibits Owner.
Customer and Bank play the role of Owners for Cash.
Transaction exhibits Type, which can be withdrawal or deposit, and Amount.
Customer handles Transaction Querying and Cash Accepting.
Transaction Querying occurs if Approval is obtained.
Transaction Querying yields Transaction.
Credit Limit Checking occurs if Type of Transaction is withdrawal.
Credit Limit Checking requires Card Data, Bank, and Amount.
Credit Limit Checking determines whether Credit Limit is exceeded.
Denial Notifying occurs if Credit Limit is exceeded, otherwise Cash Dispensing occurs.
Denial Notifying requires Card Data.
Denial Notifying yields Denial Notice.
Cash Dispensing occurs if Type of Transaction is withdrawal and if Credit Limit is not exceeded.

Credit Limit Checking and Cash Dispensing occur if Type is withdrawal.
Cash Dispensing requires Card Data and Amount.
Cash Accepting and Cash Dispensing affect Account.
Cash Dispensing changes Ownership of Cash from bank to customer.
Cash Accepting occurs if Type of Transaction is deposit.
Cash Accepting requires Card Data and Amount.
Cash Accepting changes Owner of Cash from customer to bank.
References


References


Cook, S. Foreword to Warmer and Kleppe 1999.


D’Souza, D., Sane, A. and Birchenough, A. UML Profiles Considered Redundant. 1999. home.earthlink.net/~salhir/TheUML-TwoYearsAfterAdoptionOfTheStandard.PDF


Davidson, E.J. Jerry’s Discussion Icebreakers. iscd@uindy.edu mailing list, 1998.


Evans, A. Dependencies and Associations. Precise UML Mailing List, June 8, 2001, puml-list@cs.york.ac.uk


Holmevik, J.R. The History of Simula, Center for Technology and Society, University of Trondheim, N-7055 Dragvoll, Norway, 1995.


References


Simons, T. Dependencies and Associations. puml-list@cs.york.ac.uk email forum, June 21, 2001


Bold page numbers refer to definitions. Bold italics refer to OPL sentence types.

A
abbreviated participation constraint, 116
abstract thing, 198
abstracting phase, 299
Action Data Flow Diagram, 403, 406
Activity, 408
Activity diagram, 409
ADISSA, 401
adjacency matrix, 276
adjective, 199, 200
adjective-induced shortcut, 200
adverb, 201
affected object, 95
affectee, 96
agent, 94
agent condition link, 336
Agent condition sentence, 100
agent link, 299
Agent sentence, 40, 120
aggregation complexity, 144, 198
Aggregation sentence, 37, 135
aggregation symbol, 8
Aggregation-exhibition unfolding sentence, 227
Aggregation-exhibition-generalization-classification unfolding sentence, 227
Aggregation-Participation, 35, 166, 352, 361
Aggregation-Participation hierarchy, 361
Agile Modeling, 413
algorithm, 4, 67
allopoietic system, 387, 399
ambiguity, 274
analysis, 309, 401
analytic approach, 383
analyzing, 298
ancestor, 176
anti-symmetric, 110
architecture, 249, 257
Aristotelian method, 396
Aristotle, 383, 395, 396
arrowhead, 15, 97
artificial system, 18, 256, 260, 262, 289, 390
Aspect-Oriented Programming, 412
association, 15, 61, 78, 105, 106, 129, 283, 324, 349, 407, 415, 437
atomic state, 329
atomic thing, 144
attribute, 147, 167
attribute feasibility matrix, 332
attribute mode, 167
attribute size, 167
attribute touch, 167
attribute value, 199
automated application generation, 35
autopoietic system, 387, 399
Axiomatic Design, 268
axiomatic logic, 397
B
backward direction, 108
backward relation, 107
Bacon, 397
behavior, 403
beneficiary, 389, 400
bidirectional arrowhead, 16
bidirectional structural link, 109
binary structural relation, 106
biological system, 384
biology, 384
Booch method, 407
Boolean condition sentence, 50
bottom-up, 403
bottom-up direction, 173
brain, VIII, 3, 4, 10, 39, 56, 72, 263, 283, 392
business management, 388
business opportunity, 290

C
cardinality, 117
Cartesian product, 343
Cascade which sentence, 359
causality, 395
cause and effect, 93, 388, 395
change, 84
Change sentence, 89
Characterization sentence, 145
chemistry, 279
chronon, 342
class, 193
Class Description, 408
Class Diagram, 407
classification complexity, 198
Classification unfolding sentence, 226
Collaboration diagram, 409
colorful thing, 148
Common Warehouse Metadata, 410
complex problem, 212
complexity, 70, 244, 385
complexity management, 414
Component diagram, 409
compound condition, 334
Compound condition sentence, 50
compound state, 329, 337
Compound state enumerated exhibition sentence, 165
compound thing, 144
Compound-or state enumeration sentence, 333
comprehensive fork link, 124
comprehensiveness, 124
comprehensiveness-clarity tradeoff, 88
computer evolution, 381
computer integrated manufacturing, 4
computer programming, 326
computer programming language, 7
concatenation, 36
concept, 261
contextual framework, 395
contextual graph, 382
concrete thing, 198
concurrency, 404
condition link, 23, 24, 45, 100, 334, 336
Condition sentence, 45
conjecture, 417
consolidated set, 230
consolidating, 230, 244
consolidator, 230
construction, 91
consumee, 96
consuming, 5, 10, 29, 61, 97
consumption, 91
consumption link, 97
Consumption sentence, 120
Context sensitivity, 9
contravariance, 364
count theories, 385
controlling, 390, 400
covariance, 364
creating, 7, 65, 71
customer, 4, 35, 208, 264, 295, 309, 413
cybernetics, 384
D
DAG, 350
Darwin, 384
data, 379
Data Dictionary, 408
Data Flow Diagram, 16, 401, 402, 409
database schema, 4, 34, 35, 117, 276, 308, 422
declaration, 326
decomposition principle, 212, 417
default attribute value convention, 362
default scenario, 23
default state, 335
Default state specification sentence, 314
deployment, 4, 290, 291, 294, 409
Deployment diagram, 409
Descartes, 383, 396, 397
design, 309
destroying, 7, 71
destruction, 83
detail level, 244
Determination sentence, 49
deterministic OPM, 348
DFD, see Data Flow Diagram
DFD/ERD approach, 403
diagramming method, 281
diagramming technique, 281
directed acyclic graph, 21, 26, 241, 276, 349, 350
distributive law, 25, 120, 121, 122, 129, 136, 244
distributive law of structural relations, 120
divide and conquer, 212
domain expert, 16, 34, 298
dot operator, 356
drill down, 71, 267, 289, 424
dual-state object, 50
dull thing, 148
dynamic aspect, 313
dynamic model, 404
dynamics, 256, 399

E
economic system, 384
economics, 384
effect, 84, 87
effect link, 89, 97
Effect sentence, 41, 89
elaboration, 244
electromagnetic theory, 393
electron, 397
electronic commerce, 4, 33
electronic economy, 3
elementary particle, 395
ellipse, 16, 19, 24, 25, 29, 42, 46, 223, 234, 236, 278, 282, 331
emergent feature, 165
empiricism, 381, 397
enabler, 17, 93, 94, 99, 100, 202, 203
enabling link, 95
encapsulation, 414
encapsulation principle, 415
energy, 396
engineering, 292
English grammar, 174
Enlightening, 397
enterprise, 294
Entity-Relationship Diagrams, 402
enumerated textual values, 326
environment, 384, 389, 400
Equivalence sentence, 112
essence, 66
event, 342, 374
event model, 407
event probability, 375
event trace diagram, 407
evolution, 389
exception handling, 371
executable code generation, 35
exhibition complexity, 148, 198
Exhibition process unfolding sentence, 226
Exhibition sentence, 64, 86, 145, 161, 203
Exhibition unfolding sentence, 226
Exhibition-Characterization, 144, 151, 166, 167
existence, 76, 256, 302
existence-impacting process, 314
experimental evidence, 397
experimentation, 383
expertise, 380
expressive power, 34, 39, 52, 66, 145, 274, 284
Extreme Programming, 413

F
feature, 146
feedback, 385, 400
feedback loop, 390
feedback mechanism, 388
field, 150
finance, 388
flow of control, 22, 23, 339, 341, 374
fold consolidating, 231
fork, 121
fork degree, 123
Fork tagged structure sentence, 121
form, 260
formal ontology, 383
formality, 4, 35, 276, 281
Forrester, 388
forward direction, 108
forward relation, 107
fractal relation, 363, 375
Fractal sentence, 364
framework, VII, IX, 4, 77, 91, 134, 382, 384
function, 166, 249, 251, 256, 403
function box, 268
function hierarchy, 268
Function sentence, 270
functional model, 404
functionality, 271, 290
fundamental structural relation, 144
Fusion method, 407

G
Galileo, 383, 396
General Systems Theory, 293, 383, 399
generalization complexity, 187, 198
Generalization sentence, 173
Generalization unfolding sentence, 226
Generalization-Specialization, 171, 191, 194
generation, 83
generic event, 342
generic feedback mechanism, 391
gerund, 71, 72, 153, 155, 202, 221, 398
goal, 255
goal-directed behavior, 385
goods, 266
graphic symbol overloading, 334
graphic user interface, 308
graphical construct, 9
graphics-text equivalence principle, 39
GST, see General Systems Theory

H
hamiltonian distance, 362, 375
handle, 121
hard attribute, 164
hardware, 3, 68, 74, 95, 290, 300, 302, 307, 308, 407
Hawthorn experiment, 397
Hegel, 381
Heidegger, 381
Heisenberg’s uncertainty principle, 397
heterogeneous structural relation, 226
hierarchy, 166
homogeneous structural relation, 226
human factors, 399
human factors engineering, 294
human interface, 308
human resource management, 388
Hume, 397
identifiability, 56
identity, 24, 34, 56, 59, 67, 74, 75, 76, 77, 78, 91, 311, 314, 389, 393, 395, 396
image understanding, 4
Implementation diagrams, 409
implicit attribute, 158
In-diagram in-zooming sentence, 185, 335
industrial engineering, 397
informal ontology, 382
informatic, 380
informatical object, 57, 59, 66, 67, 72, 73, 74, 75, 156, 341, 398, 399
informatic system, 56
informatics hierarchy, 379, 399
information, 379
information system, X, 67, 68, 73, 165, 260, 304, 381, 402, 424
information theory, 385
ingenious, 380
inheritance, 403
Inheritance Graph, 408
Initial and ultimate state specification sentence, 313
initial state, 335
initiation, 4, 291, 294
input link, 7, 9, 11, 17, 87, 88, 90, 92, 106, 232, 278, 370
input state, 7, 60, 87, 240, 346, 375
instance, 191, 193, 361
instance qualification, 196
instance qualification link, 336
Instance qualification sentence, 196, 200, 316
instantiation, 191
Instantiation sentence, 191, 194
instrument, 17, 40, 41, 95, 197, 238, 245, 256, 342
instrument link, 299
Instrument sentence, 41
intent, 249, 255
interaction diagram, 407, 409
interdisciplinary knowledge, 293
interface, 6, 17, 35, 94, 238, 276, 302, 307, 379, 404
interface model, 408
interim detail level, 244
intermediate object, 361
Internet, 3, 130, 438
introspection, 383
invocation link, 97
involved object set, 99
In-zooming sentence, 43, 69, 140
K
Kant, 55, 133, 381, 438
Keynes, 384
knowledge, 379
Knowledge Interchange Format, 382
L
legend, 5, 282
Leibniz, 397
lifecycle, 313
lifecycle cost, 291
lifecycle expression, 408
lifecycle model, 408
lifecycle support, 294
linearity, 274
linguistic object, 401
Listing sentence, 239, 241
locality, 395
Locke, 397
logical AND, 24, 25, 42, 46, 333, 334, 335
logical compound state, 332
logical OR, 25, 46, 47, 333, 334
logical XOR, 25, 46, 331, 371
look-and-feel, 308
<table>
<thead>
<tr>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>maintainability, 291</td>
<td>natural language, 34, 325, 349, 396</td>
</tr>
<tr>
<td>manifestation, 244</td>
<td>natural language construct, 283</td>
</tr>
<tr>
<td>manufacturability, 291</td>
<td>natural language prose, 290</td>
</tr>
<tr>
<td>manufacturing, 388</td>
<td>natural sciences, 292</td>
</tr>
<tr>
<td>marketing, 388</td>
<td>Negative Boolean condition sentence, 50</td>
</tr>
<tr>
<td>mathematical function, 276, 277</td>
<td>New diagram selective semi-folding sentence, 241</td>
</tr>
<tr>
<td>mathematical modeling, 276</td>
<td>New diagram semi-folding sentence, 239</td>
</tr>
<tr>
<td>mathematics, 292</td>
<td>Newton, 393</td>
</tr>
<tr>
<td>maximal teeth set, 123</td>
<td>Newton’s first law, 59</td>
</tr>
<tr>
<td>Measurement unit specification sentence, 160</td>
<td>Nietzsche, 381</td>
</tr>
<tr>
<td>mechanics theory, 393</td>
<td>non-comprehensiveness, 203, 204</td>
</tr>
<tr>
<td>memory, 399</td>
<td>non-comprehensiveness symbol, 142</td>
</tr>
<tr>
<td>mental model, 381</td>
<td>non-reserved OPL phrase, 373</td>
</tr>
<tr>
<td>mental record, 56, 399</td>
<td>non-reserved phrase, 37</td>
</tr>
<tr>
<td>metamodel approach, 410</td>
<td>noun, 57, 60, 61, 79, 155, 157, 174, 198</td>
</tr>
<tr>
<td>metamodeling level, 312</td>
<td>null tag, 111</td>
</tr>
<tr>
<td>Meta-Object Facility, 410, 418</td>
<td>numeral, 319</td>
</tr>
<tr>
<td>meta-OPD, 285</td>
<td></td>
</tr>
<tr>
<td>meta-OPM specification, 369</td>
<td></td>
</tr>
<tr>
<td>metaphysical revolution, 396</td>
<td></td>
</tr>
<tr>
<td>methodology, 3, 4, 5, 207, 208, 271,</td>
<td></td>
</tr>
<tr>
<td>283, 289, 410, 417, 422</td>
<td></td>
</tr>
<tr>
<td>middle-out analysis, 49, 244, 293, 299</td>
<td></td>
</tr>
<tr>
<td>mode, 164</td>
<td></td>
</tr>
<tr>
<td>model, 272</td>
<td></td>
</tr>
<tr>
<td>model integration, 20</td>
<td></td>
</tr>
<tr>
<td>model multiplicity, 414</td>
<td></td>
</tr>
<tr>
<td>model multiplicity problem, 416</td>
<td></td>
</tr>
<tr>
<td>Model-Driven Architecture, 410</td>
<td></td>
</tr>
<tr>
<td>modeling, 272, 382</td>
<td></td>
</tr>
<tr>
<td>modeling level, 283</td>
<td></td>
</tr>
<tr>
<td>modeling methodology, 281</td>
<td></td>
</tr>
<tr>
<td>modifier, 199</td>
<td></td>
</tr>
<tr>
<td>Module diagram, 407</td>
<td></td>
</tr>
<tr>
<td>MOSES, 407</td>
<td></td>
</tr>
<tr>
<td>multiple inheritance, 189</td>
<td></td>
</tr>
<tr>
<td>multiplicity constraint, 15, 36, 114</td>
<td>objectifying, 399, 400</td>
</tr>
<tr>
<td>object, 57</td>
<td></td>
</tr>
<tr>
<td>Object Behavior Analysis, 406</td>
<td></td>
</tr>
<tr>
<td>object dependability, 112</td>
<td></td>
</tr>
<tr>
<td>Object Diagram, 407</td>
<td></td>
</tr>
<tr>
<td>Object Interaction Graph, 408</td>
<td></td>
</tr>
<tr>
<td>Object Life-Cycle, 406</td>
<td></td>
</tr>
<tr>
<td>Object Management Group, 408</td>
<td></td>
</tr>
<tr>
<td>object model, 404</td>
<td></td>
</tr>
<tr>
<td>Object Modeling Technique, 404</td>
<td></td>
</tr>
<tr>
<td>Object paradigm, 149, 403, 414</td>
<td></td>
</tr>
<tr>
<td>Object Structure Analysis, 406</td>
<td></td>
</tr>
<tr>
<td>object symbol, 13</td>
<td></td>
</tr>
<tr>
<td>Object technology, X, 402</td>
<td></td>
</tr>
<tr>
<td>object transformation, 112</td>
<td></td>
</tr>
<tr>
<td>Objectchart, 407</td>
<td></td>
</tr>
<tr>
<td>object-class diagram, 19</td>
<td></td>
</tr>
<tr>
<td>object-class model, 16</td>
<td></td>
</tr>
<tr>
<td>Object-Flow Diagram, 403</td>
<td></td>
</tr>
</tbody>
</table>
Object-Oriented, X, 13, 77
Object-Oriented Analysis, 405
Object-Oriented Design, 405
Object-Oriented methods, 19, 93
Object-Oriented Process,
  Environment and Notation, 408
Object-Oriented programming, 76
Object-Oriented Software
  Engineering, 404
Object-Oriented Systems Analysis,
  405
Object-Process Diagram, X, 5, 13, 29,
  33, 35, 52, 106, 373, 374, 439, 441
Object-Process Language, 5, 6, 10,
  33, 35, 52
object-process paradigm, 61
of chain, 353
OMG, 408
OML, 408
OMT, 404
ontological inquiry, 382
ontology, 289, 382, 399
OO method, 16, 55, 403, 407, 413,
  416
OPD, see Object-Process Diagram
OPD construct, 374
OPD item, 53
OPD label, 26, 242
OPD sentence, 109, 373, 374
OPD set, 35, 123, 373, 374
OPD symbol, 10, 13, 142, 159, 374
OPD-OPL pair, 53
OPEN, 408
OPEN Modeling Language, 408
open system, 384
operand, 101
operating system, 326
operation, 147, 167, 360, 375
operation model, 408
operator, 101, 408
OPL, see Object-Process Language
OPL construct, 374
OPL item, 53
OPL paragraph, 38, 373, 374
OPL phrase, 373, 374
OPL script, 373, 374
OPL sentence, 373, 374
OPM construct hierarchy, 366
OPM construct pair, 374
OPM meta-model, 284
OPM paragraph pair, 374
OPM phrase pair, 374
OPM sentence pair, 373, 374
OPM specification, 373, 374
OPM system process, 308
Optional use sentence, 120
OR result sentence, 47
organism, 384, 389
organized complexity, 385
organized simplicity, 385
origin, 70
output link, 7, 9, 11, 17, 88, 90, 92,
  106, 189, 231, 232, 370
output state, 7, 9, 60, 87, 88, 346, 370,
  375
Out-zooming sentence, 49

P
paradigm, 401, 415
parameterized participation
  constraint, 142
part of speech, 60, 198
partial solution, 212
participation constraint, 14, 37, 113,
  114, 115, 116, 117, 118, 119, 120,
  123, 129, 136, 142, 143
participation level, 143
Participation sentence, 135
path label, 231
Path-labeled change sentence, 233
perseverance, 64
persistent, 11, 56, 59, 62, 64, 66, 71,
  73, 129, 307
philosophy, 382
phrase, 106
physical medium, 399
physical object, 59, 67, 68, 72, 75, 398
Physical object sentence, 69
Physical process sentence, 69
physics, 393
Plato, 396
Plural process specialization sentence, 174
Plural specialization sentence, 172, 173
positive transitivity, 361
postprocess object set, 22, 61, 99, 100
precedence, 299
predicate calculus, 382
preprocess object set, 22, 61, 99, 100, 344, 398
primary operand, 244
probabilistic link, 408
probabilistic OPD, 348
problem statement, 13, 14, 15, 16, 18, 20, 48
procedure, 147
process diagram, 407
Process OR result sentence, 47
Process specialization sentence, 173
process start event, 343
process test, 61, 62, 78, 112, 398
Process XOR result sentence, 46
processing state, 346
product, 263, 264
product development process, 291
productive thing, 187
productivity, 397
program code, 4, 35
programming language, 4, 33, 34, 45, 49, 117, 147, 175, 199, 356, 403, 418
project, 267
prose, 275

Q
qualification inheritance, 188, 190
qualification link, 188, 336
Qualification sentence, 65, 188
qualitative attribute, 163
quantitative attribute, 164
quantum mechanics, 396
quantum theory, 393, 397, 400
query, 204, 356

R
range participation constraint, 115
Rational Unified Process, 412
rationalism, 381, 397
real-time, 68, 339, 417, 422
real-time feedback, 34
reciprocal structural relation, 110
Reciprocal structure sentence, 110
reciprocity, 111
rectangle, 6, 13, 14, 25, 85, 204, 228, 229, 235, 282
Recursive Design, 406
reduction-oxidation reaction, 279
refinement, 20, 27, 28, 49, 71, 89, 141, 210, 211, 290, 418
refining phase, 299
reflective metamodeling, 162, 284, 286
reflective thinking, 388
Relation sentence, 37, 112
relational database, 150
relativistic effect, 395
relativity, 254
relativity theory, 393
religion, 397
requirement elicitation, 4
reserved object, 159, 167
reserved OPL phrase, 373
reserved phrase, 9, 37, 44, 45, 46, 47, 52, 85, 116, 128, 135, 140, 142, 354
reserved word of, 375
reserved word which, 357, 375
Respective agent sentence, 180
Respective instrument sentence, 179, 180
result, 8, 9, 11, 15, 17, 18, 42, 92, 97, 186, 237, 305, 341
Result sentence, 42
resultee, 96
reverse engineering, 259
robot, 381
*Role-playing sentence*, 315
routangl, 6, 85, 160, 161, 231
routine, 147

**S**

science, 397
scientific analysis, 383, 385
scientific method, 292, 383
scope, 351, 360
secondary operand, 244
selective semi-folding, 240
self-identity, 395, 396
semantic sentence analysis, 60
semantics, 9, 283
semi-folded object, 239
semi-folding, 239, 244
semi-unfolding, 244
separation of concerns, 416
sequence diagram, 409
services, 266
Shannon, 385
side effect, 360
Sight Code, X, 34, 54, 209, 419, 420, 440
Simula, 414
simulation, 348
Single value sentence, 322
*Sink “which sentence”,* 359
soft attribute, 164
software, 407
software engineer, X
software engineering, 291
software process, 291
software system, 289, 291
solipsism, 397
source, 302
*Source “which” sentence*, 359
specialization, 7
*Specialization sentence*, 172, 194, 203

specialization-specification hierarchy, 365
SQL statement, 356
stand-alone process, 19
state, 83
state duration, 375
state entrance event, 343
*State enumerated exhibition sentence*, 64, 86, 165
*State enumeration sentence*, 44, 64, 85, 86, 163, 319, 323
state expression, 89
state phrase, 318
*State specialization sentence*, 182, 183
*State specification sentence*, 317
state suppression, 89
State Transition Diagram, 407
Statecharts, 404, 407
state-impacting process, 314
state-maintaining link, 317
*State-maintaining sentence*, 317
state-specified consumption link, 312, 336
*State-specified consumption sentence*, 184, 185, 312
*State-specified generation sentence*, 50
state-specified object, 311, 335
state-specified result link, 312, 336
*State-specified result sentence*, 184, 312
State-specified transformation sentence, 313
static/dynamic view, 407
sterile thing, 187
stochastic process, 348
storage requirement, 326
strategy, 212, 261, 404
strong transitive structural relation, 362
structure, 403

**Structure sentence**, 36, 38, 110, 140

structure-behavior combination, 261

structured design, 401

subjective attribute, 255

subjective mental world, 396

subjectivity, 254

supply-chain design, 291

symbol, 13

symbolic value, 326

symmetry, 358

synergy principle, 383

syntax, 34, 35, 37, 60, 74, 78, 114, 182, 196, 233, 251, 283, 408, 417

synthesis-oriented view, 383

system, 253

system analysis paradigm, 413

system architect, 258, 294, 354, 417

system architecture, 261

system developer, 3, 148

system development, VII, 4, 207, 208, 290, 294, 299, 388, 417

system development method, 212

system diagram, 18, 19, 21, 24, 26, 28, 29, 38, 40, 43, 44, 54, 71, 245, 376, 426

system dynamics, 83, 384

system evolution, 291

system generation, 309

system integrator, 294

system lifecycle, 289

system map, 21, 26, 28, 30, 43, 51, 229, 241, 242, 244, 245, 425

Systemantica, X, 34, 54, 209, 420, 421, 422

systems analysis, 291

systems engineer, 294

systems theory, 379

T

table column, 356

tagged structural link, 15, 107, 113, 129

tagged structural relation, 357

**Tagged structure sentence**, 36, 121, 122

Task, 408

Technique, 408

teeth set, 123

termination, 4, 91, 341, 343

textual value, 325

thermodynamic theory, 393

thermodynamics, 384, 396

ting, 62

Thing symbol, 145

time exception handling, 371

timeline, 22, 23, 61, 112, 138, 155, 156, 339, 342

tooth, 121

top-down analysis, 244

top-down direction, 173

top-down specification, 49

top-level system diagram, 21

transform, 101

transformation link, 89, 96

transformee, 96

transforming link, 93, 96, 202, 237

transitive relation, 173

transitive structural relation, 125, 194

transitivity, 125, 361

transitivity strength, 361, 375

triggering event, 342

U

Ultimate OPD, 27

ultimate state, 335

UML, 136, 143, 408

understanding, 380

unidirectional structural link, 109

Unified Modeling Language, 408

**Unit specified value enumeration sentence**, 160

unorganized complexity, 385

use case, 94

use-case diagram, 409

user interface, 308

V

value, 63, 324

**Value effect sentence**, 161
Value enumeration sentence, 161, 319, 323
value phrase, 318
Value role-playing sentence, 316
Value sentence, 318
Value specification sentence, 317
Value-specified exhibition sentence, 322
visibility, 244
Visibility Graph, 408

W
weak transitive structural relation, 362
Weaver, 385
whole-part relation, 8, 14, 139, 142
Wiener, 385
wisdom, 380

X
XOR effect-instrument sentence, 236
XOR instrument sentence, 236
XOR result sentence, 42, 46
XOR source change sentence, 319

Z
zoom consolidating, 231
zooming in, 43
zooming out, 28
**Procedural Links**

These links are generally used between an object and a process. They cannot be used to link objects together.

<table>
<thead>
<tr>
<th>Link Name</th>
<th>OPD Symbol</th>
<th>OPL Sentence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumption</td>
<td>![Image]</td>
<td>Processing consumes Object.</td>
<td>Process uses object up entirely during its occurrence.</td>
</tr>
<tr>
<td>Effect</td>
<td>![Image]</td>
<td>Processing affects Object.</td>
<td>Process changes the state of the object in an unspecified manner.</td>
</tr>
<tr>
<td>Input and Output</td>
<td>![Diagram]</td>
<td>Processing changes Object from input state to output state.</td>
<td>The object is at input state prior to the process occurrence, and at output state as a result of its occurrence.</td>
</tr>
<tr>
<td>Agent</td>
<td>![Image]</td>
<td>Object handles Processing.</td>
<td>Object is a human that is not changed by the process; process needs the agent object in order to occur.</td>
</tr>
<tr>
<td>Instrument</td>
<td>![Image]</td>
<td>Processing requires Object.</td>
<td>Object is a non-human that is not changed by the process; process needs the instrument object in order to occur.</td>
</tr>
<tr>
<td>Invocation</td>
<td>![Image]</td>
<td>X Processing invokes Y Processing.</td>
<td>First process directly starts up a second process, without an inter-mediate object.</td>
</tr>
</tbody>
</table>

**States**

*State sentences and images*

```
| Object | state 1 | state 2 | state 3 |
```

Object can be state 1, state 2, or state 3.

```
| Object | Value 1 | Value 2 | Value 3 |
```

Values of Object are Value 1, Value 2, and Value 3.
### States (continued)

#### State-related Links

<table>
<thead>
<tr>
<th>Link Name</th>
<th>OPD Symbol</th>
<th>OPL Sentence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Condition</strong></td>
<td><img src="image" alt="Processing" /></td>
<td>Processing occurs if Object is state 1.</td>
<td>Object is an instrument. It must be at a specific state in order for the process to occur.</td>
</tr>
<tr>
<td><strong>Agent Condition</strong></td>
<td><img src="image" alt="Object" /> <img src="image" alt="Processing" /></td>
<td>Object must be at state 2 for Processing to occur.</td>
<td>Object is an agent. It must be at a specific state in order for the process to occur.</td>
</tr>
<tr>
<td><strong>Qualification</strong></td>
<td><img src="image" alt="Object" /> <img src="image" alt="state 1" /> <img src="image" alt="Attribute" /> <img src="image" alt="Qualified Object" /></td>
<td>Qualified Object is an Object, the Attribute of which is state 1.</td>
<td>Qualified Object is a type of Object. It must be at a particular state of Object’s Attribute.</td>
</tr>
<tr>
<td><strong>Instance Qualification</strong></td>
<td><img src="image" alt="Object" /> <img src="image" alt="state 1" /> <img src="image" alt="Attribute" /> <img src="image" alt="Qualified Object" /></td>
<td>Qualified Object is an instance of an Object, the Attribute of which is state 1.</td>
<td>Qualified Object is an instance of class Object. It must be at a particular state of Object’s Attribute.</td>
</tr>
<tr>
<td><strong>State Specified Consumption</strong></td>
<td><img src="image" alt="Object" /> <img src="image" alt="state" /> <img src="image" alt="Processing" /></td>
<td>Processing consumes state Object.</td>
<td>Process consumes object only if it is at a certain state.</td>
</tr>
<tr>
<td><strong>State Specified Result</strong></td>
<td><img src="image" alt="Object" /> <img src="image" alt="state" /> <img src="image" alt="Processing" /></td>
<td>Processing yields state Object.</td>
<td>Process creates object at a certain state.</td>
</tr>
</tbody>
</table>

#### Boolean Objects

Specialized informatical objects. Boolean objects are questions, and they always have two states (the answers): yes and no.

<table>
<thead>
<tr>
<th>Link type</th>
<th>OPD Symbols</th>
<th>OPL Sentence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Determination</strong></td>
<td><img src="image" alt="Determining" /></td>
<td>Determining determines whether Object is proper.</td>
<td>Process yields a Boolean object that poses a “yes or no” question. The process then determines the answer.</td>
</tr>
<tr>
<td><strong>Condition link</strong></td>
<td><img src="image" alt="Object is proper?" /> <img src="image" alt="yes" /> <img src="image" alt="no" /> <img src="image" alt="Processing" /></td>
<td>A Processing occurs if Object is proper. B Processing occurs if Object is not proper.</td>
<td>If the answer is “yes,” a certain process occurs. If the answer is “no”, a different process occurs.</td>
</tr>
<tr>
<td><strong>Negative condition link</strong></td>
<td><img src="image" alt="Object is proper?" /> <img src="image" alt="yes" /> <img src="image" alt="no" /> <img src="image" alt="Processing" /></td>
<td>A Processing occurs if Object is proper, otherwise B Processing occurs.</td>
<td>Compound sentence: if the answer is “yes,” a certain process occurs, otherwise a different process occurs.</td>
</tr>
<tr>
<td><strong>Both condition links</strong></td>
<td><img src="image" alt="Object is proper?" /> <img src="image" alt="yes" /> <img src="image" alt="no" /> <img src="image" alt="Processing" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>