

INDEX

μ C/OS-II, 99

A

abstract

- channel, 30, 31
- data, 30, 31
- task, 30, 31

abstract channel, 30, 31, 33

abstract data, 30, 31

abstract data items, 143

abstract declarations, 30

abstract task, 32, 75

all-data task, 47

annotations, 30

any-data task, 47

application composition, 165

architecture

- Blackboard, 14

- Harvard, 26

architecture-independent programming, 175

ATaG, 12

- Abstract Task Graph, 16

ATaG behavior, 112, 119

ATaG programs

- libraries, 109

ATaG simulator, 121, 127

AtagManager, 82

attribute-based task placement, 37

automatic software synthesis, 117

B

Blackboard architecture, 14

C

case study, 135

ChannelDeclaration, 72

channels

- input, 112

- output, 112

code placement, 34

code skeletons, 114, 119

communication, 12

generative, 12
 orthogonality, 12
 compilation, 115
 lazy, 101
 component-based design, 67
 composability, 109
 computing
 Data-Driven Computing, 23
 context labels, 176
 control-driven program flow, 23
 coordinate system, 5
 coordination language
 Linda, 25

D

DART, 17, 65
 Data-Driven ATaG Runtime, 17
 data fusion, hierarchical, 55
 data-driven ATaG Runtime, 17
 data-driven computing, 23, 176
 data-driven program flow, 23
 dataflow programs, 24
 Datapool, 79
 declarations
 abstract, 30
 declarative programming, 28
 demand-driven programming, 26
 density-based instantiation, 36
 dependent task, 46
 Dispatcher, 90
 Distributed Oz, 12
 DOSBART, 14

E

event-driven processing, 24

F

Firing Rule, 32
 Firing rules, 46
 framework for compilation, 177
 functional programming, 26

G

GME, 105, 106
 Generic Modeling Environment, 105,
 106, 142, 178

get(), 48, 97, 98, 178
 graphical interface, 105, 128
 guarantee of "correctness", 116

H

Harvard architecture, 26
 hierarchical data fusion, 55

I

imperative, 23
 imperative programming, 28
 in-network
 aggregation, 52
 processing, 6
 input channels, 112
 Instantiation, 32
 interpreters
 model, 106

J

Java virtual machine, 102
 JavaSpaces, 25

K

Kairos, 9, 15

L

language
 macroprogramming, 177
 layers of programming abstraction, 7
 lazy compilation, 101
 leader election, 139
 libraries of ATaG programs, 109
 Linda, 12, 25

M

macroprogramming, 8, 14, 177
 MIC, 106
 Model Integrated Computing, 106
 model interpreter, 119
 model interpreters, 106, 133

N

neighbor-to-neighbor interaction, 52
 nesC, 11
 network description, 113
 NetworkArchitecture, 88
 NetworkStack, 87
 node-centric programming, 10, 14

O

object tracking, 50, 51, 127, 139, 170
 output channels, 112

P

periodic task, 47
 platform independence, 66
 priority assignment, 102
 processing

- event-driven, 24
- reactive, 26

 program flow

- data-driven, 23

 programming

- architecture-independent, 175
- declarative, 28
- demand-driven, 26
- functional, 26
- imperative, 28
- macroprogramming, 8, 14
- node-centric, 10, 14
- reduction, 26

 programming abstraction

- layers, 7

 programming languages

- Imperative, 23

 programming style

- Regiment, 10

 put(), 48, 97, 98, 178

R

reactive processing, 26
 reactive system, 3
 reduction programming, 26
 Regiment, 15
 Regiment programming style, 10
 resource annotations, 38

resource management, 61
 run(), 152, 162
 Runnable interface, 148

S

sensor network, 1
 service-oriented specification, 7
 situatedness, 88, 123
 software synthesis, 171
 space awareness, 3

T

task, 45

- abstract, 75
- all-data, 47
- any-data, 47
- dependent, 46
- periodic, 47
- scheduling, 102

 task graph, 29
 task invocation, 34
 task placement, 34
 task scheduling, 102
 TaskDeclaration, 72
 temperature gradient monitoring, 136, 137
 TinyDB, 15
 transformational systems, 3
 tuple space, 12
 tuple spaces, 25

U

utility-based negotiator, 63

V

virtual topology, 52
 visual programming environment, 112
 visualization, 129
 visualizer package, 129, 132

- GUI, 129

 von Neumann machine, 3

W

Wireless sensor networks, 1

