APPENDIX 2c

Entity Relationship concordance for Core Concepts in the Software Design Minimal Methods

Action (Variable). [n015]; StateVector has . . .

Action is_a FunctionName. [n16]

After. [n166]; IntervalConstraint may_be . . .

Allocation has FunctionalMappings (List). [n125]

Allocation is_a Method. [n124]

Articulation has DataDictionary (List). [n081]

Articulation has Situation. [n079]

Articulation is_a Method. [n078]

Assignment. ; Function has atleast one . . .

Before. [n165]; IntervalConstraint may_be . . .

Bubble has ControlArc (Relation). [n060]

Bubble has ControlSpec (Holder). [n061]

Bubble has InputDataArcs (Parameters). [n065]

Bubble has Mode. [n058]

Bubble has OutputDataArcs (Parameters). [n066]

Bubble is Triggered. [n067]; When InputDataArcs present . . .

Bubbles (Function). [n054]; Context has . . .

Bubbles (List). [n057]; Bubbles decompose_into . . .

Bubbles decompose_into Bubbles (List). [n057]

Bubbles. [n062]; DataArcs maps_to . . .

Bunch has Intervals (List). [n176]

Bundle has Signals (List). [n173]

Bundles (Holder). [n161]; Sheaf has . . .

Color (Type). [n041]; Markers has . . .

CommunicationChannel has CommunicationMechanism. [n104]

CommunicaitonChannel has DataArcs. [n111]

CommunicationChannel has Protocol. [n110]

CommunicationChannel. [n119]; Task receives . . .; Message from . . CommunicationChannel

CommunicationMechanism may_be Flag. [n108]

CommunicationMechanism may_be Queue. [n105]

CommunicationMechanism may_be Rendezvous. [n106]

CommunicationMechanism may_be Semaphore. [n107]

CommunicationMechanism may_be Variable. [n109]

CommunicationMechanism. [n104]; CommunicationChannel has . . .

CommunicationsChannels (List). [n094]; DARTS has . . .

ConcurrentDesign has ProcessingElements. [n097]

ConcurrentDesign. [n094]; DARTS has . . .

Context (Bubble). [n053]; Decomposition has . . .

Context has Bubbles (Function). [n054]

ControlArc (Relation). [n060]; Bubble has . . .

ControlSpec (Holder). [n061]; Bubble has . . .

ControlSpec establishes Mode. [n072]

ControlSpec may_have DecisionTable. [n069]

ControlSpec may_have PetriNet. [n071]

ControlSpec may_have ProcessActivationTable. [n070]

ControlSpec may_have StateMachine. [n068]

CurrentState (Variable). [n006]; StateMachine has . . .

CurrentState maps_to NowCurrentState. [n019]

CurrentState. [n018]; NextState maps_to . . .

DARTS has CommunicationsChannels (List). [n094]

DARTS has ConcurrentDesign. [n094]

DARTS has DistributedDesign. [n093]

DARTS has Monitors. [n116]

DARTS is_a method. [n092]

DataArcs (Relation). [n055]; Decomposition has . . .

DataArcs maps_to Bubbles. [n062]

DataArcs maps_to DataStore. [n063]

DataArcs maps_to Externals. [n064]

DataArcs. [n111]; CommunicaitonChannel has . . .

DataDictionary (List). [n081]; Articulation has . . .

DataDictionary has Entity. [n082]

DataDictionary has RelationArc (Relation). [n083]

DataItem (Variable). [n059]; DataItems has . . .

DataItems (List). [n088]; DataStores has . . .

DataItems has DataItem (Variable). [n059]

DataItems has DataTypes (Type). [n090]

DataItems. [n089]; Operations modify . . .

DataItems. [n113]; Message has . . .

DataStore (Holder). [n056]; Decomposition has . . .

DataStore (Holder). [n086]; Entity has . . .

DataStore. [n063]; DataArcs maps_to . . .

DataStore. [n117]; Monitor has . . .

DataStores has DataItems (List). [n088]

DataTypes (Type). [n090]; DataItems has . . .

Datum maps_to Variables. [n157]

Datum. [n152]; InformationFlow has . . .

DecisionTable. [n069]; ControlSpec may_have . . .

Decomposition has Context (Bubble). [n053]

Decomposition has DataArcs (Relation). [n055]

Decomposition has DataStore (Holder). [n056]

Decomposition has Externals (source or sink). [n052]

Decomposition has Mode (Type). [n051]

Decomposition is_a Method. [n050]

DesignElement Actions. [n147]; System Actions maps_to. . .

DesignElement Actions maps_to System States. [n148]

DesignElement States. [n149]; System Actions maps_to . . .

DesignElement.States. [n146]; System.States maps_to . . .

DesignElementFlow has DesignElements. [n141]

DesignElementFlow has System. [n142]

DesignElementFlow is_a Method. [n140]

DesignElements. [n141]; DesignElementFlow has . . .

DesignElment has StateMachine. [n145]

DiachronicMapping. [n154]; InformationFlow has . . .

DistributedDesign has ProcessingElements. [n096]

DistributedDesign. [n093]; DARTS has . . .

Dual (Variable). [n004]; Method may_have . . .

Duration (Variable). [n177]; Interval has . . .

During. [n167]; IntervalConstraint may_be . . .

Entities (List). [n085]; Entity decomposes_into . . .

Entities from Entities. [n084]; RelationArcs map_to . . .

Entities with RelationArcs. [n080]; Situation has . . .

Entity decomposes_into Entities (List). [n085]

Entity has DataStore (Holder). [n086]

Entity has Operations (Function). [n087]

Entity. [n082]; DataDictionary has . . .

Equals. [n172]; IntervalConstraint may_be . . .

Equation. [n075]; Function may_have . . .

Event (String). [n178]; Interval has . . .

Event (Variable). [n012]; StateVector has . . .

Event maps_to SMInput. [n020]

ExecutiveLoop. [n123]; Task has . . .

Externals (source or sink). [n052]; Decomposition has . . .

Externals. [n064]; DataArcs maps_to . . .

Finishes. [n169]; IntervalConstraint may_be . . .

FirePetriNet(null) is_a Operation. [n049]

Flag. [n108]; CommunicationMechanism may_be . . .

FromViewPoint (Variable). [n002]; Method has . . .

Function has atleast one Assignment.

Function maps_to ProcessingElements. [n121]

Function may_have Equation. [n075]

Function may_have Loop. [n073]

Function may_have Rule. [n076]

Function may_have Selector. [n074]

Function. [n120]; Task has . . .

Function. [n122]; Task has . . .; Selector of . . .

Function. [n135]; Instruction is_a . . .

FunctionName maps_to SMOutput. [n021]

FunctionName. ; Action is_a . . .

FunctionalArc (List of Relations). [n129]; FunctionalMapping has . . .

FunctionalArc maps_to ProcessingElement from Function. [n130]

FunctionalMapping depends_on SystemMode (Variable). [n126]

FunctionalMapping has FunctionalArc (List of Relations). [n129]

FunctionalMapping has Functions (List). [n127]

FunctionalMapping has ProcessingElements (List). [n128]

FunctionalMappings (List). [n125]; Allocation has . . .

Functions (List). [n127]; FunctionalMapping has . . .

GetAction(IN.SMInput->.Event;.OUT.SMOutput->.Action) is_a Operation. [n027]

GetCurrentState(OUT CurrentState) is_a Operation. [n024]

In Diachronic mapping the datum over time moves through a set of variables. [n156]

In Synchronic mapping the values in a set of variables relate to a single timespan. [n155]

InformationFlow has Datum. [n152]

InformationFlow has DiachronicMapping. [n154]

InformationFlow has SynchronicMapping. [n153]

InformationFlow has Variables. [n151]

InformationFlow is_a Method. [n150]

InputDataArcs (Parameters). [n065]; Bubble has . . .

InputPlaces (List). [n043]; Transit has . . .

Instruction is_a Function. [n135]

Instructions. [n133]; VirtualMachine has . . .

Interval (List). [n174]; Signal has . . .

Interval decomposes_into Intervals (List). [n180]

Interval has Duration (Variable). [n177]

Interval has Event (String). [n178]

Interval has State (String). [n179]

IntervalConstraint may_be After. [n166]

IntervalConstraint may_be Before. [n165]

IntervalConstraint may_be During. [n167]

IntervalConstraint may_be Equals. [n172]

IntervalConstraint may_be Finishes. [n169]

IntervalConstraint may_be Meets. [n171]

IntervalConstraint may_be Overlapping. [n170]

IntervalConstraint may_be Starts. [n168]

IntervalConstraints. [n164]; SignalArc has . . .

Intervals (List). [n176]; Bunch has . . .

Intervals (List). [n180]; Interval decomposes_into . . .

Lacune (List). [n175]; Signal has . . .

LeftHandSide has Marker Colors in InputPlaces. [n046]

LeftHandSide. [n044]; PetriRule has . . .

Loop. [n073]; Function may_have . . .

Marker Colors in InputPlaces. [n046]; LeftHandSide has . . .

Markers (List). [n033]; Petrinet has . . .

Markers has Color (Type). [n041]

Markers is_a Tokens (Type). [n037]

Markers propagate_along PetriArcs. [n040]

Meets. [n171]; IntervalConstraint may_be . . .

Message from CommunicationChannel. [n119]; Task receives . . .

Message has DataItems. [n113]

Messages. [n112]; Protocol has . . .

Messages.(List) associated_with_one ProcessingElement. [n137]; Worldline has . . .

Messages.(List) between ProcessingElements. zn39] ; Scenario has_causally_related . . .

Method has FromViewPoint (Variable). [n002]

Method has Name (Variable). [n001]

Method has ToViewPoint (Variable). [n003]

Method may_have Dual (Variable). [n004]

Method. [n092]; DARTS is_a . . .

Method. [n005]; StateMachine is_a . . .

Method. [n028]; Petrinet is_a . . .

Method. [n050]; Decomposition is_a . . .

Method. [n078]; Articulation is_a . . .

Method. [n124]; Allocation is_a . . .

Method. [n131]; VirtualMachine is_a . . .

Method. [n136]; WorldLine is_a . . .

Method. [n138]; Scenario is_a . . .

Method. [n140]; DesignElementFlow is_a . . .

Method. [n150]; InformationFlow is_a . . .

Method. [n159]; Temporality is_a . . .

Mode (Type). [n051]; Decomposition has . . .

Mode (Variable). [n008]; StateMachine has . . .

Mode maps_to StateVectors. [n022]

Mode. [n058]; Bubble has . . .

Mode. [n072]; ControlSpec establishes . . .

Monitor has DataStore. [n117]

Monitor has Semaphore. [n118]

Monitors. [n116]; DARTS has . . .

Name (Variable). [n001]; Method has . . .

NextState (Variable). [n014]; StateVector has . . .

NextState maps_to CurrentState. [n018]

NowCurrentState (Variable). [n013]; StateVector has . . .

NowCurrentState. [n019]; CurrentState maps_to . . .

Operation. [n024]; GetCurrentState(OUT CurrentState) is_a . . .

Operation. [n025]; SetInitialState(IN.CurrentState) is_a . . .

Operation. [n026]; SetInitialVectorList(IN.StateVectors) is_a . . .

Operation. [n027]; GetAction(IN.SMInput->.Event;.OUT.SMOutput->.Action) is_a . . .

Operation. [n049]; FirePetriNet(null) is_a . . .

Operations (Function List). [n023]; StateMachine has . . .

Operations (Function). [n087]; Entity has . . .

Operations (List). [n048]; Petrinet has . . .

Operations modify DataItems. [n089]

OutputDataArcs (Parameters). [n066]; Bubble has . . .

Overlapping. [n170]; IntervalConstraint may_be . . .

PetriArc (Relation) from Place to Transit. [n036]; PetriArcs has . . .

PetriArcs (List). [n032]; PetriMatrix has . . .

PetriArcs has PetriArc (Relation) from Place to Transit. [n036]

PetriArcs maps_to Places. [n038]

PetriArcs maps_to Transits. [n039]

PetriArcs. [n040]; Markers propagate_along . . .

PetriMatrix has PetriArcs (List). [n032]

PetriMatrix has Places (List). [n030]

PetriMatrix has Transits (List). [n031]

PetriMatrix. [n029]; Petrinet has . . .

PetriNet. [n071]; ControlSpec may_have . . .

PetriRule has LeftHandSide. [n044]

PetriRule has RightHandSide. [n045]

PetriRules (List). [n042]; Transit has . . .

Petrinet has Markers (List). [n033]

Petrinet has Operations (List). [n048]

Petrinet has PetriMatrix. [n029]

Petrinet is_a Method. [n028]

Place (Variable). [n034]; Places has . . .

Places (List). [n030]; PetriMatrix has . . .

Places has Place (Variable). [n034]

Places. [n038]; PetriArcs maps_to . . .

ProcessActivationTable. [n070]; ControlSpec may_have . . .

ProcessingElement from Function. [n130]; FunctionalArc maps_to . . .

ProcessingElement. [n098]; ProcessorArrays is_a . . .

ProcessingElement. [n101]; Task is_a . . .

ProcessingElement. [n137]; Worldline has . . .; Messages.(List) associated_with_one . . .

ProcessingElements (List). [n128]; FunctionalMapping has . . .

ProcessingElements. [n096]; DistributedDesign has . . .

ProcessingElements. [n097]; ConcurrentDesign has . . .

ProcessingElements. [n121]; Function maps_to . . .

ProcessingElements. zn39] ; Scenario has_causally_related . . .; Messages.(List) between . . .

ProcessorArrays decompose_into ProcessorArrays. [n099]

ProcessorArrays has Processors. [n100]

ProcessorArrays is_a ProcessingElement. [n098]

ProcessorArrays. [n099]; ProcessorArrays decompose_into . . .

Processors has Tasks. [n102]

Processors. [n100]; ProcessorArrays has . . .

Protocol has Messages. [n112]

Protocol has Receiver StateMachine. [n115]

Protocol has Sender StateMachine. [n114]

Protocol. [n110]; CommunicationChannel has . . .

Queue. [n105]; CommunicationMechanism may_be . . .

Receiver StateMachine. [n115]; Protocol has . . .

RelationArc (Relation). [n083]; DataDictionary has . . .

RelationArc has RelationAttributes (Variable). [n091]

RelationArcs map_to Entities from Entities. [n084]

RelationArcs. [n080]; Situation has . . .; Entities with . . .

RelationAttributes (Variable). [n091]; RelationArc has . . .

Rendezvous. [n106]; CommunicationMechanism may_be . . .

RightHandSide triggers Transit. [n047]

RightHandSide. [n045]; PetriRule has . . .

Rule. [n076]; Function may_have . . .

SMInput (Parameter). [n009]; StateMachine has . . .

SMInput. [n020]; Event maps_to . . .

SMOutput (Parameter). [n010]; StateMachine has . . .

SMOutput. [n021]; FunctionName maps_to . . .

Scenario has_causally_related Messages (List) between ProcessingElements.

Scenario is_a Method. [n138]

Selector of Function. [n122]; Task has . . .

Selector. [n074]; Function may_have . . .

Semaphore. [n107]; CommunicationMechanism may_be . . .

Semaphore. [n118]; Monitor has . . .

Sender StateMachine. [n114]; Protocol has . . .

SetInitialState(IN.CurrentState) is_a Operation. [n025]

SetInitialVectorList(IN.StateVectors) is_a Operation. [n026]

Sheaf (Holder). [n160]; Temporality has . . .

Sheaf has Bundles (Holder). [n161]

Sheaf has SignalArc (Relation). [n162]

Signal from Signal. [n163]; SignalArc maps_to . . .

Signal has Interval (List). [n174]

Signal has Lacune (List). [n175]

SignalArc (Relation). [n162]; Sheaf has . . .

SignalArc has IntervalConstraints. [n164]

SignalArc maps_to Signal from Signal. [n163]

Signals (List). [n173]; Bundle has . . .

Situation has Entities with RelationArcs. [n080]

Situation. [n079]; Articulation has . . .

Starts. [n168]; IntervalConstraint may_be . . .

State (String). [n179]; Interval has . . .

StateMachine has CurrentState (Variable). [n006]

StateMachine has Mode (Variable). [n008]

StateMachine has Operations (Function List). [n023]

StateMachine has SMInput (Parameter). [n009]

StateMachine has SMOutput (Parameter). [n010]

StateMachine has StateVectors (List). [n007]

StateMachine is_a Method. [n005]

StateMachine. [n068]; ControlSpec may_have . . .

StateMachine. [n134]; VirtualMachine may_have . . .

StateMachine. [n144]; System has . . .

StateMachine. [n145]; DesignElment has . . .

StateVector has Action (Variable). [n015]

StateVector has Event (Variable). [n012]

StateVector has NextState (Variable). [n014]

StateVector has NowCurrentState (Variable). [n013]

StateVector has Variables (Set). [n011]

StateVector is_a Trigger for Function. [n017]

StateVectors (List). [n007]; StateMachine has . . .

StateVectors. [n022]; Mode maps_to . . .

SynchronicMapping. [n153]; InformationFlow has . . .

System Actions maps_to DesignElement Actions. [n147]

System Actions maps_to DesignElement States. [n149]

System has StateMachine. [n144]

System has System. [n143]

System. [n142]; DesignElementFlow has . . .

System. [n143]; System has . . .

System States. [n148]; DesignElement Actions maps_to . . .

System.States maps_to DesignElement.States. [n146]

SystemMode (Variable). [n126]; FunctionalMapping depends_on . . .

Task has ExecutiveLoop. [n123]

Task has Function. [n120]

Task has Selector of Function. [n122]

Task is_a ProcessingElement. [n101]

Task receives Message from CommunicationChannel. [n119]

Tasks decompose_into Tasks. [n103]

Tasks. [n102]; Processors has . . .

Tasks. [n103]; Tasks decompose_into . . .

Temporality has Sheaf (Holder). [n160]

Temporality is_a Method. [n159]

ToViewPoint (Variable). [n003]; Method has . . .

Tokens (Type). [n037]; Markers is_a . . .

Transit (Function). [n035]; Transits has . . .

Transit has InputPlaces (List). [n043]

Transit has PetriRules (List). [n042]

Transit. [n047]; RightHandSide triggers . . .

Transits (List). [n031]; PetriMatrix has . . .

Transits has Transit (Function). [n035]

Transits. [n039]; PetriArcs maps_to . . .

Trigger for Function. [n017]; StateVector is_a . . .

Variable. [n109]; CommunicationMechanism may_be . . .

Variables (Set). [n011]; StateVector has . . .

Variables maps_to Variables via Datum. [n158]

Variables via Datum. [n158]; Variables maps_to . . .

Variables. [n151]; InformationFlow has . . .

Variables. [n157]; Datum maps_to . . .

VirtualMachine decomposes_into VirtualMachines. [n132]

VirtualMachine has Instructions. [n133]

VirtualMachine is_a Method. [n131]

VirtualMachine may_have StateMachine. [n134]

VirtualMachines. [n132]; VirtualMachine decomposes_into . . .

When InputDataArcs present Bubble is Triggered. [n067]

WorldLine is_a Method. [n136]

Worldline has Messages (List) associated with one ProcessingElement. [n137]

Apeiron Press

PO Box 4402
Garden Grove,
California 92842-4402

714-638-7376
714-638-1210
palmer@think.net
palmer@netcom.com
palmer@exo.com
Dataline 714-638-0876

This book was set using Framemaker document publishing software by the author.

Electronic Version in Adobe Acrobat PDF available at http://server.snni.com:80/~palmer/homepage.html

Library of Congress
Cataloging in Publication Data

Palmer, Kent Duane

WILD SOFTWARE META-SYSTEMS

Bibliography
Includes Index

1. Philosophy-- Ontology
2. Software Engineering
3. Software Design Methods

I. Title

[XXX000.X00 199x]
9x-xxxxx
ISBN 0-xxx-xxxxx-x

Keywords:

Software, Design Methods, Ontology,
Integral Software Engineering
Methodology, Systems Theory