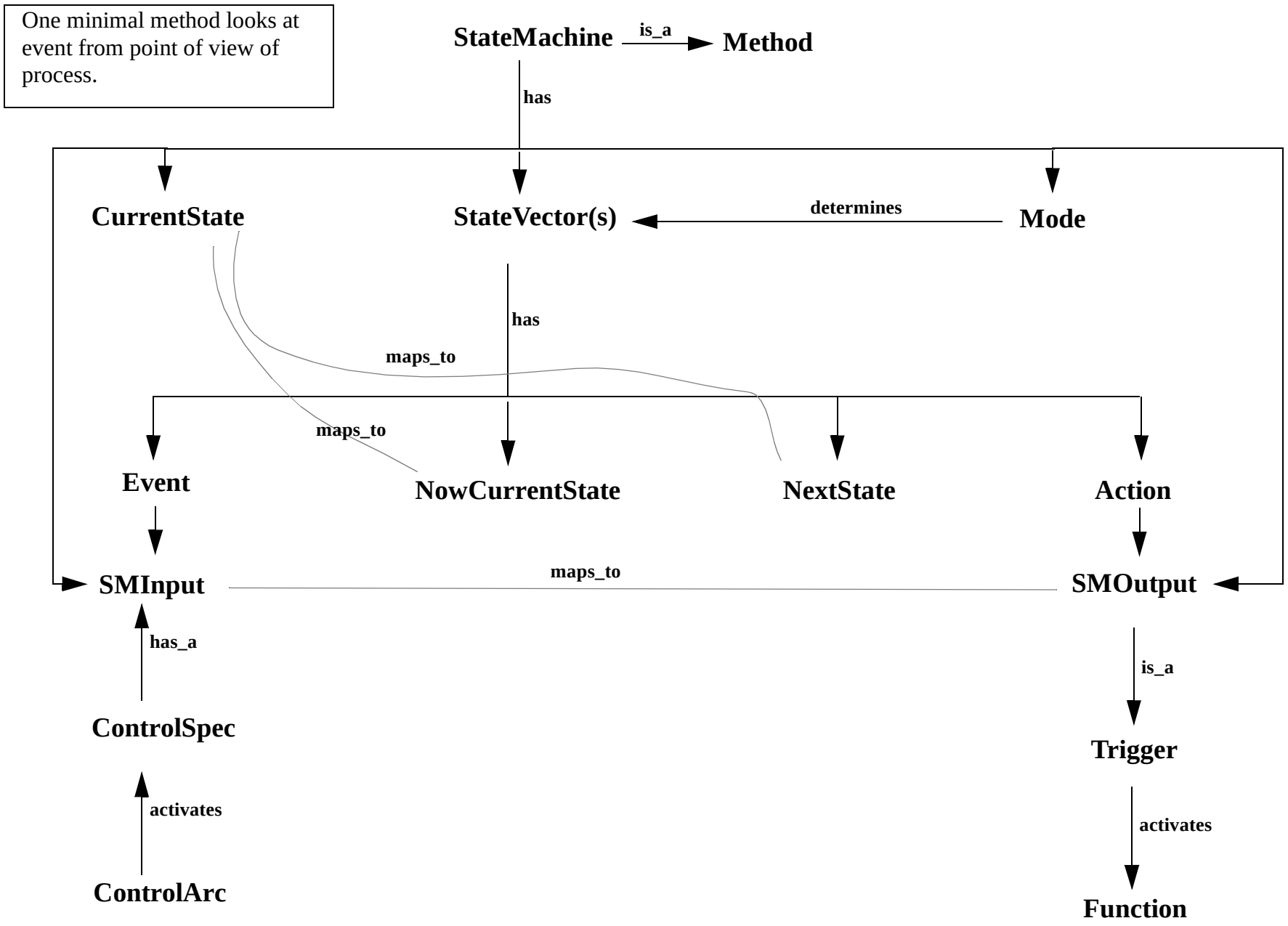
APPENDIX 2b

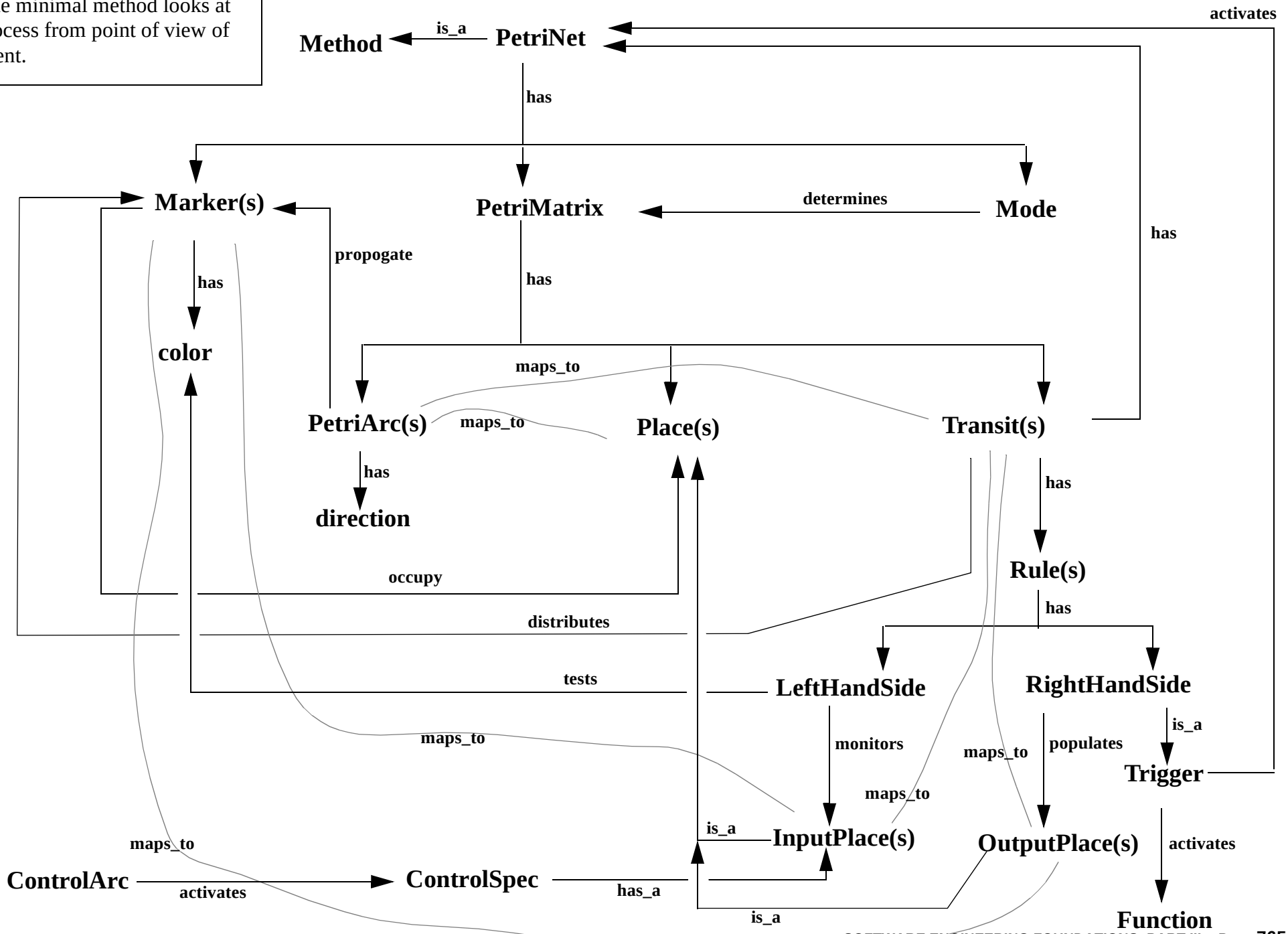
Entity Relationship diagrams

for Core Concepts in the

Software Design Minimal Methods

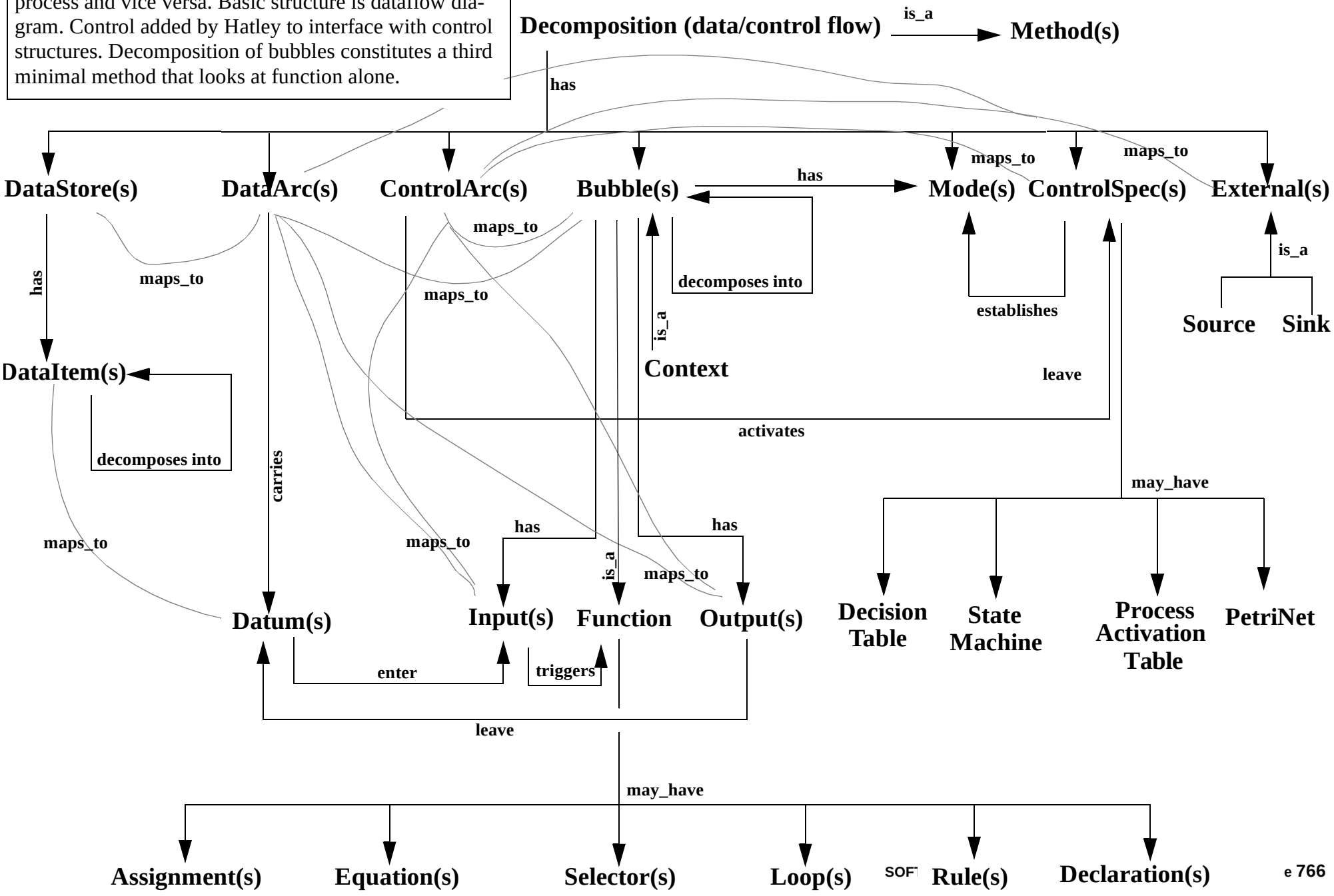

One minimal method looks at event from point of view of process.
StateMachine
is_a
Method
has
CurrentState
StateVector(s)
determines
Mode
has
maps_to
maps_to
Event
NowCurrentState
NextState
Action
SMInput
maps_to
SMOutput
has_a
is_a
ControlSpec
Trigger
activates
activates
ControlArc
Function

One minimal method looks at process from point of view of event.

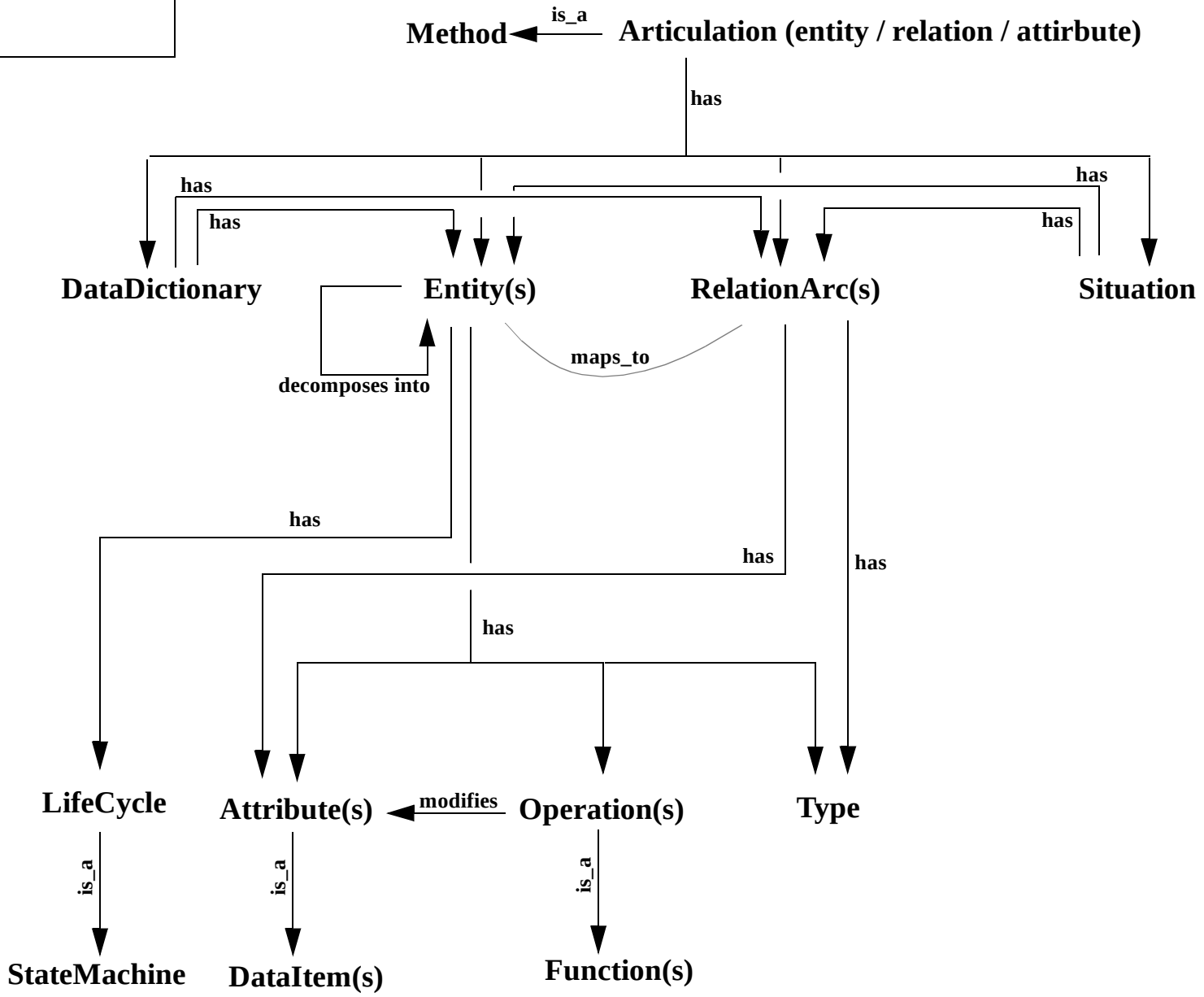


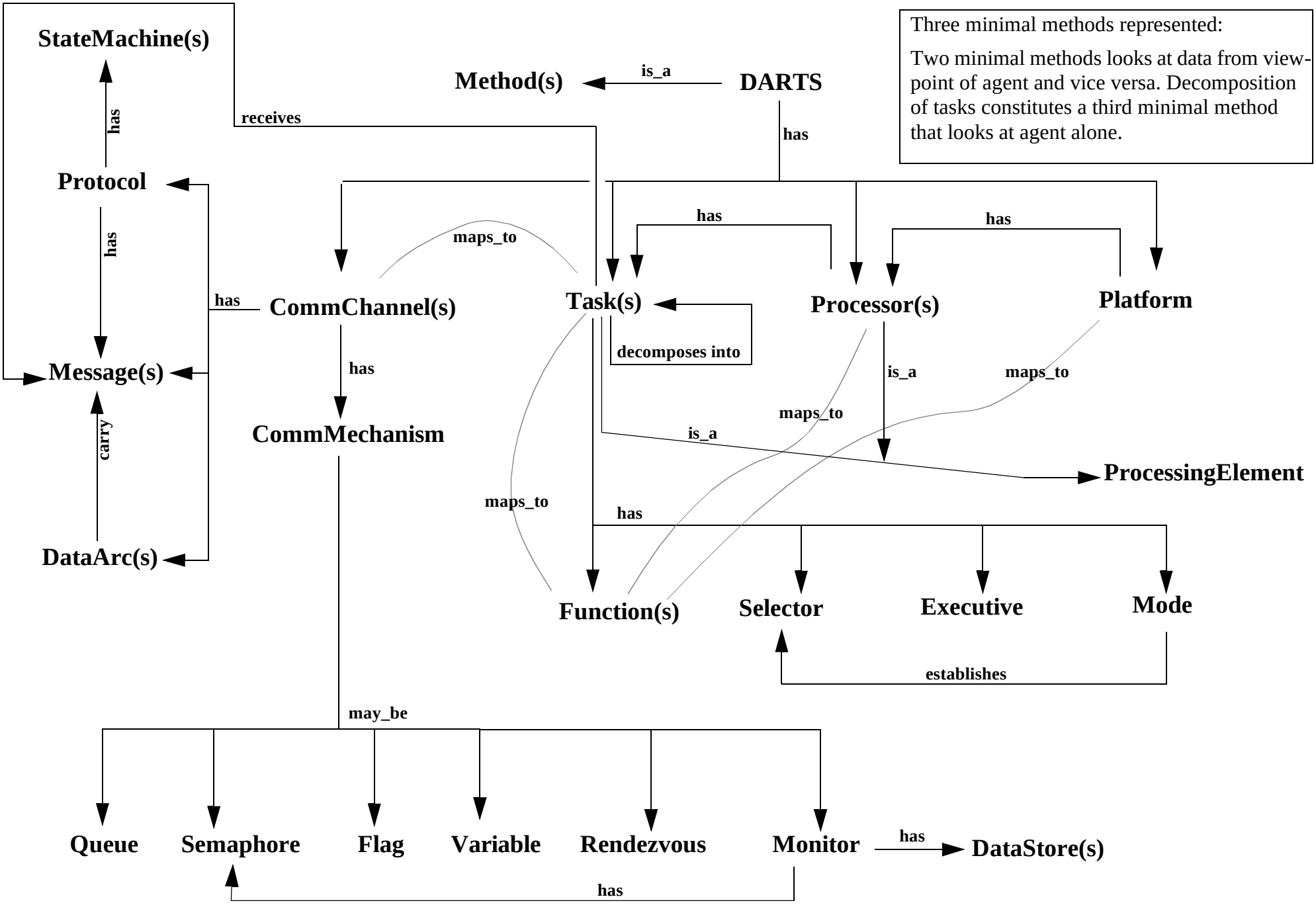
Three minimal methods represented:

Two minimal methods looks at data from viewpoint of process and vice versa. Basic structure is dataflow diagram. Control added by Hatley to interface with control structures. Decomposition of bubbles constitutes a third minimal method that looks at function alone.

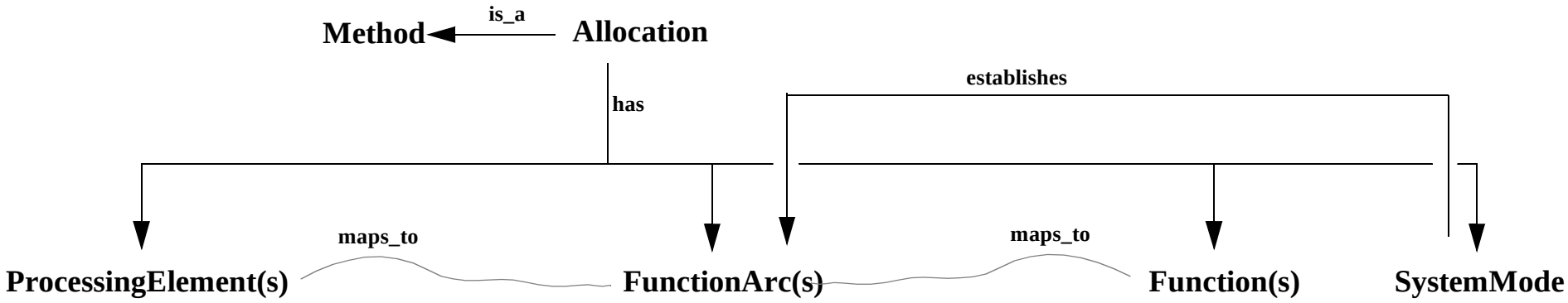


One minimal method looks data in isolation.

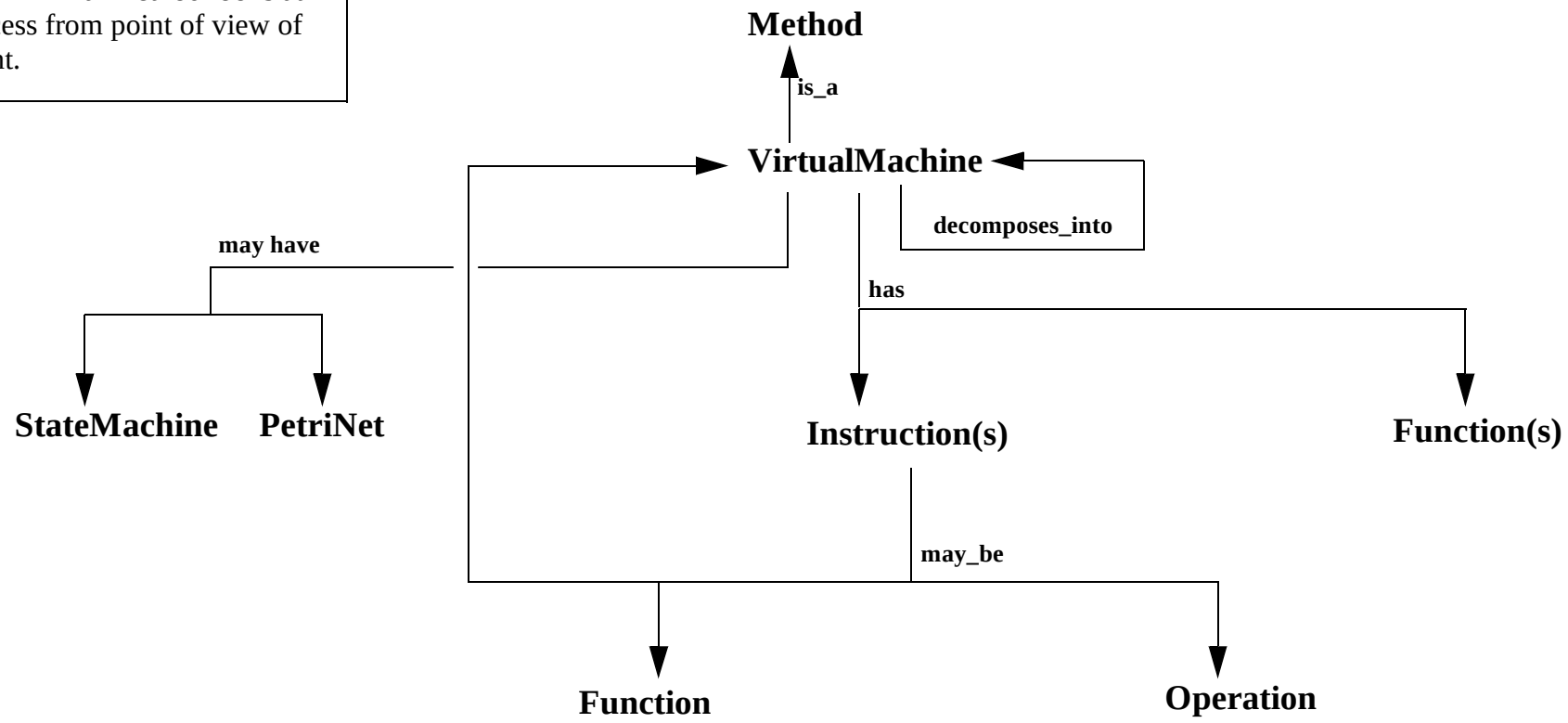



StateMachine(s)
Method(s)
is_a
DARTS
Three minimal methods represented:
Two minimal methods looks at data from viewpoint of agent and vice versa. Decomposition of tasks constitutes a third minimal method that looks at agent alone.
has
receives
has
Protocol
has
has
has
maps_to
has
CommChannel(s)
Task(s)
Processor(s)
Platform
decomposes into
Message(s)
has
is_a
maps_to
maps_to
carry
CommMechanism
is_a
ProcessingElement
maps_to
has
DataArc(s)
Function(s)
Selector
Executive
Mode
establishes
may_be
Queue
Semaphore
Flag
Variable
Rendezvous
Monitor
has
DataStore(s)
has

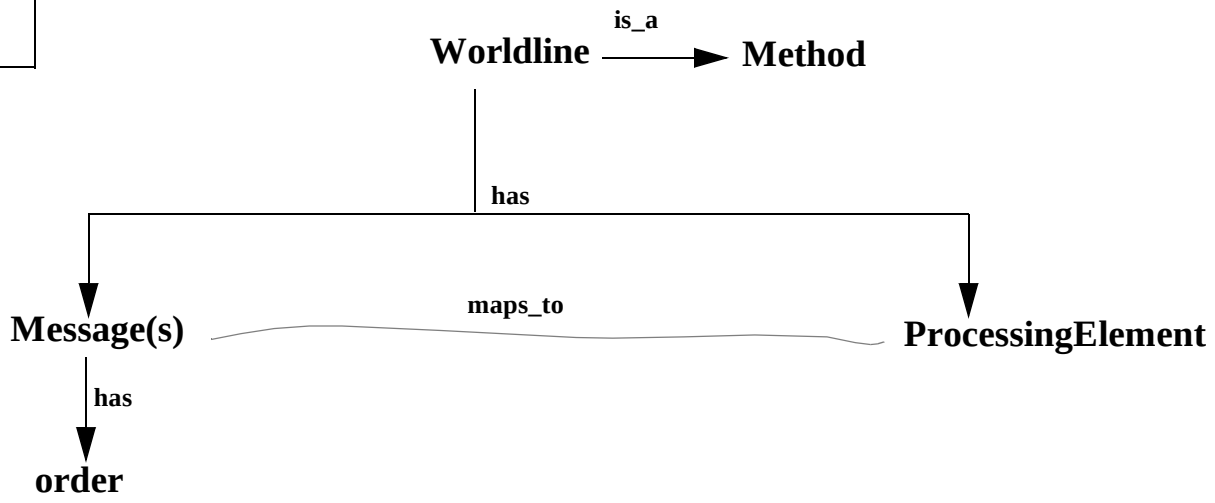
One minimal method looks at agent from point of view of process.



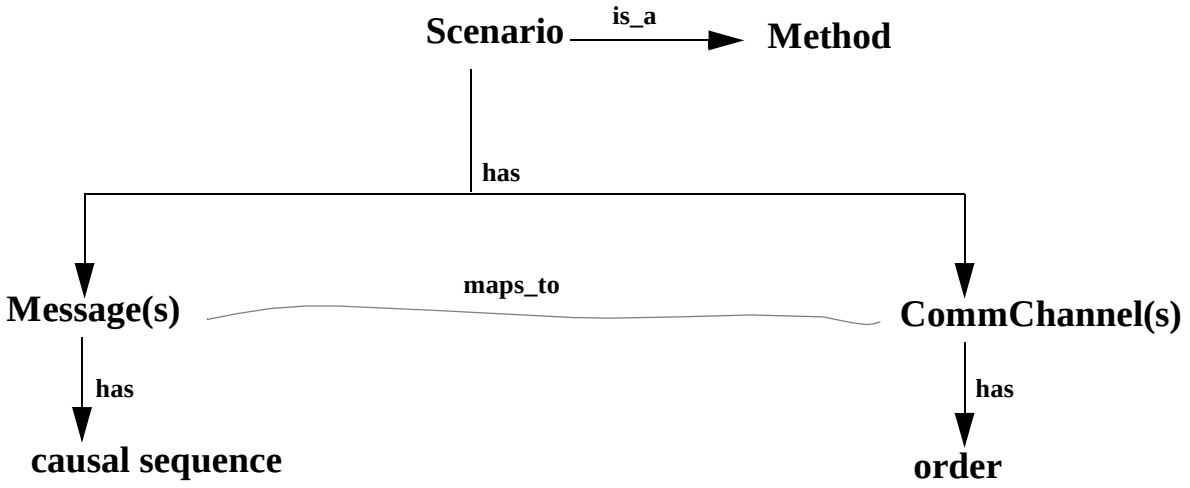
One minimal method looks at process from point of view of agent.

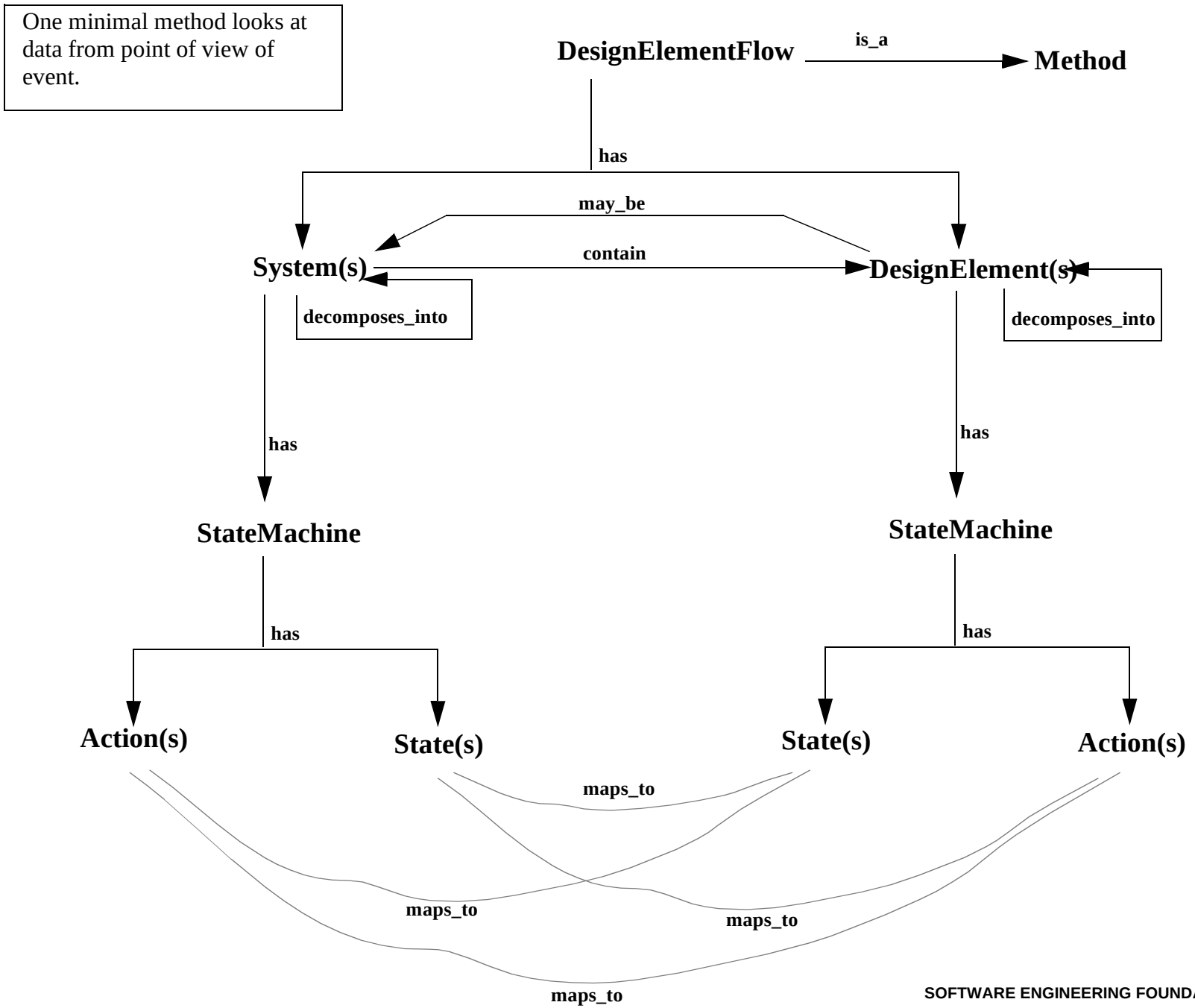


One minimal method looks at-event from point of view of agent.

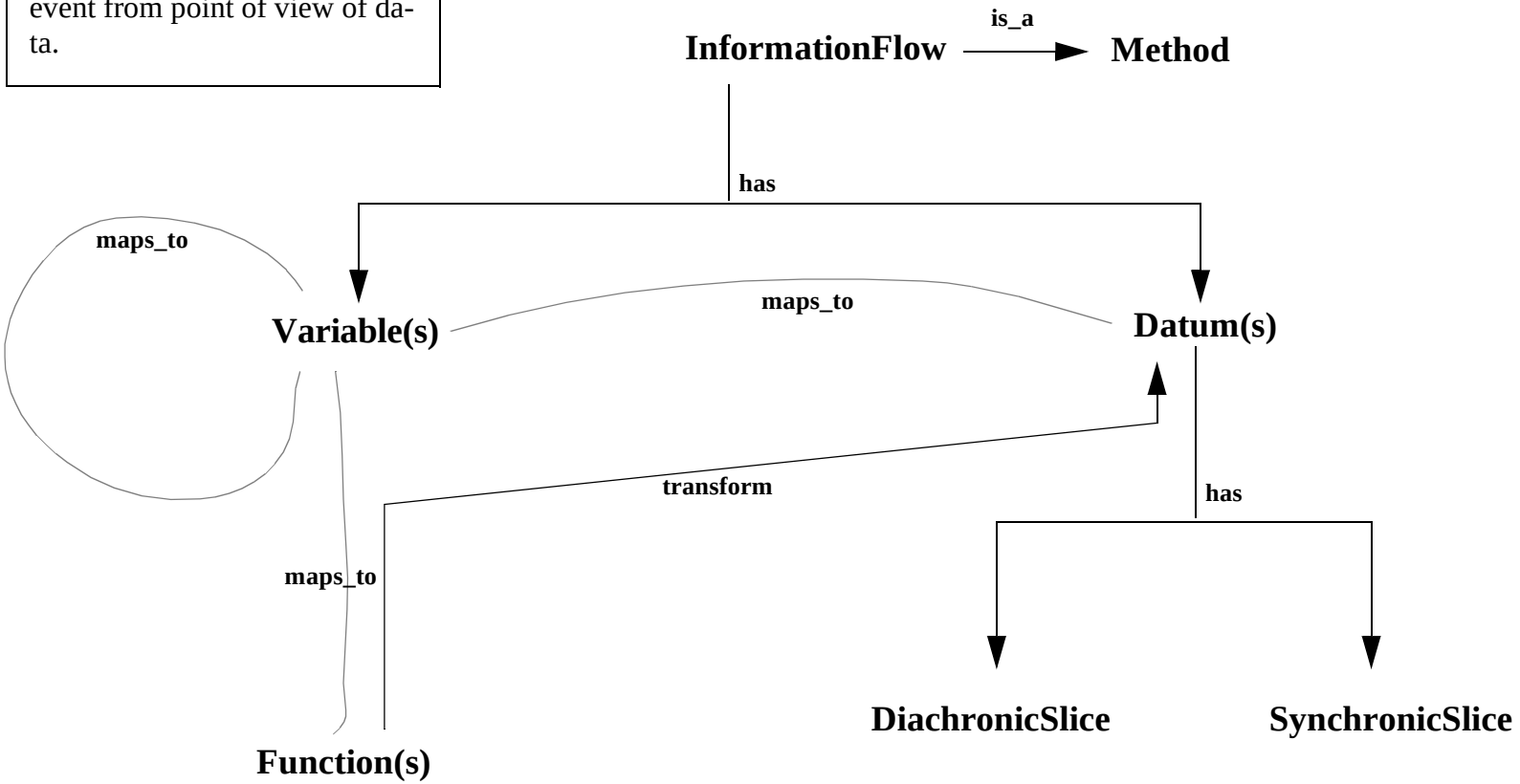


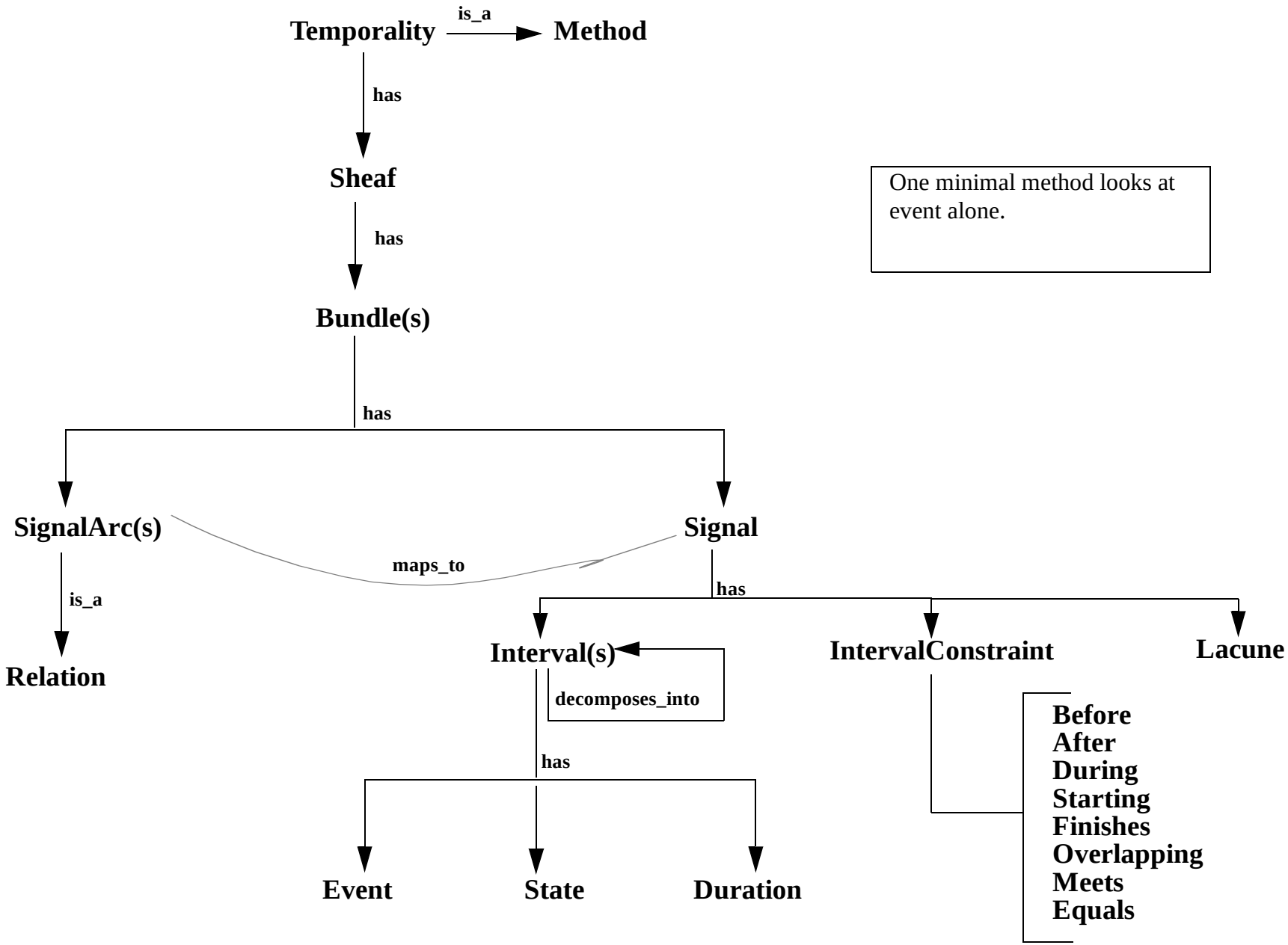
One minimal method looks at agent from point of view of event.




One minimal method looks at data from point of view of event.
DesignElementFlow
is_a
Method
has
may_be
contain
System(s)
DesignElement(s)
decomposes_into
decomposes_into
has
has
StateMachine
StateMachine
has
has
Action(s)
State(s)
State(s)
Action(s)
maps_to
maps_to
maps_to
maps_to

One minimal method looks at event from point of view of data.




Temporality
is_a
Method
has
Sheaf
has
Bundle(s)
has
SignalArc(s)
maps_to
Signal
is_a
Relation
has
Interval(s)
decomposes_into
IntervalConstraint
Lacune
has
Event
State
Duration
Before
After
During
Starting
Finishes
Overlapping
Meets
Equals
One minimal method looks at event alone.

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

Draft #1 950710 Editorial Copy.
Not for distribution.

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

Integral Software Engineering Methodology (ISEM): Entity-Relation Diagram of Core Concepts