

Hessam S. Sarjoughian
<https://acims.asu.edu>

Winter Simulation Conference
Las Vegas, NV, USA
December 3-6, 2017

© Hessam S. Sarjoughian

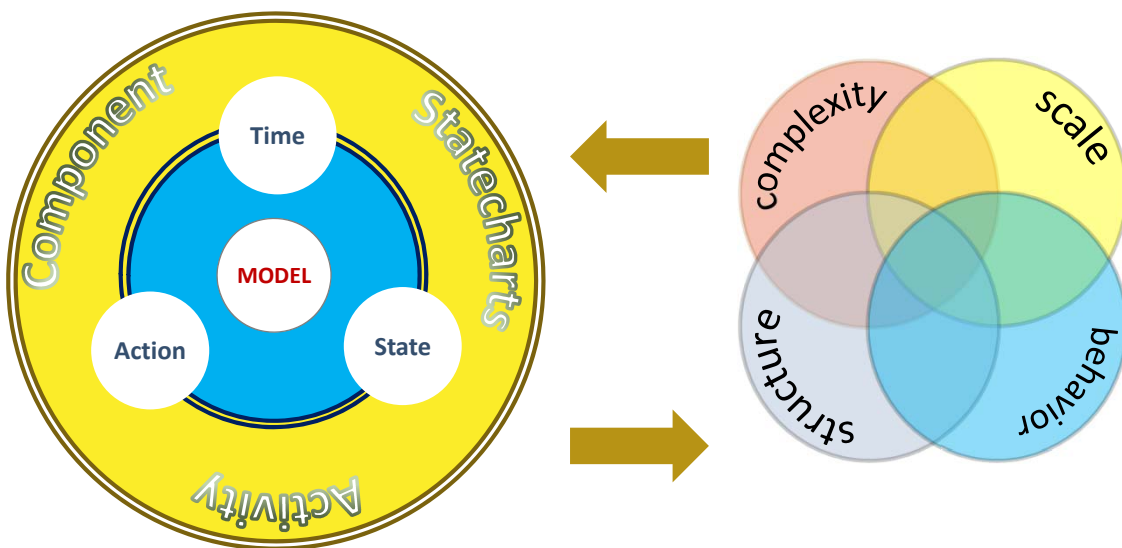
12/6/2017

Restraining *Complexity & Scale Traits* for *Component-Based Simulation Models*



<https://acims.asu.edu>

synopsis



© Hessam S. Sarjoughian

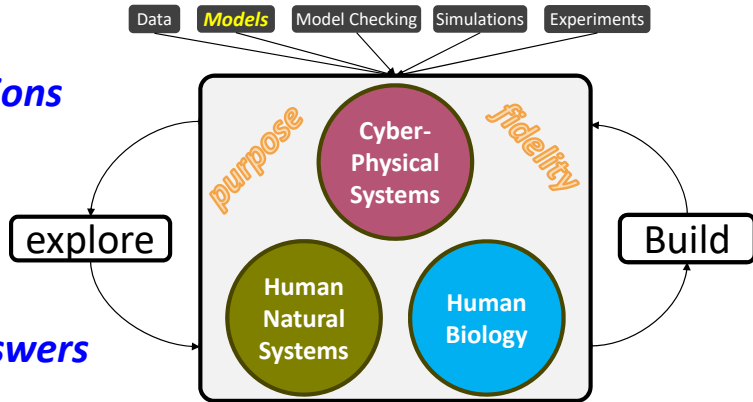
12/6/2017

model-based study of systems

models of systems are instrumental in

→ posing *useful questions*

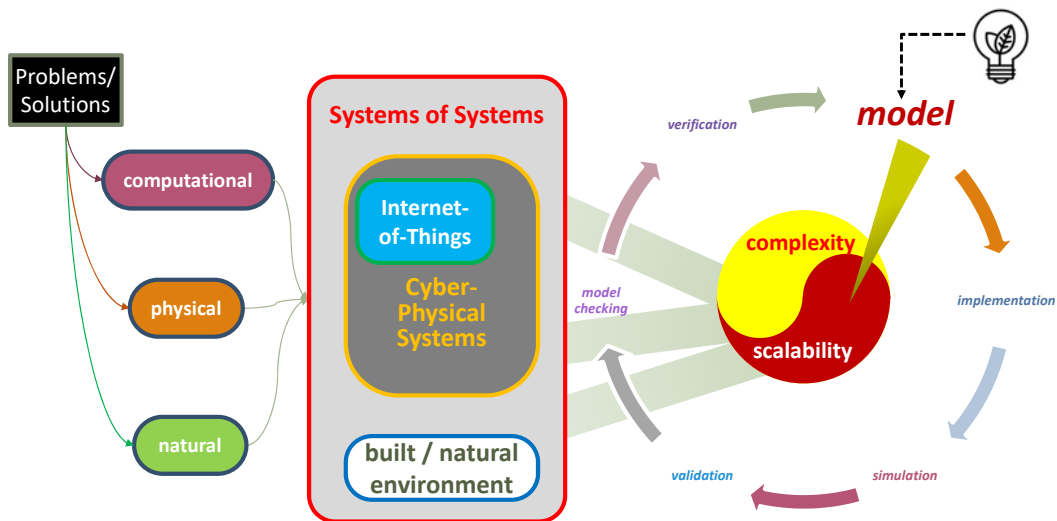
→ reaching *helpful answers*



Sztipanovits, J., X. Koutsoukos, G. Karsai, N. Kottenstette, P. Antsaklis, V. Gupta, B. Goodwine, J. Baras, and S. Wang, (2011), "Toward a Science of Cyber-Physical System Integration", Proceedings of the IEEE 100 (1): 29–44.
 Lee, E.A., (2015), "The Past, Present and Future of Cyber-Physical Systems: A Focus on Models", Sensors 15(3): 4837–4869.

<https://acims.asu.edu>

M&S lifecycle



continuous and discrete dynamical models

| Model elements | system-theoretic modeling methods | | |
|-----------------------------|-----------------------------------|----------------------|-----------------------|
| | <i>Differential Equation</i> | <i>Discrete-Time</i> | <i>Discrete-Event</i> |
| time base | continuous | discrete | continuous |
| inputs, outputs, and states | vector space | arbitrary | arbitrary |
| input segments | piecewise-continuous | sequences | discrete events |
| state and output segments | continuous | sequences | piecewise-constant |

- ➔
- implementations of model types are integrated
 - model types are *conceptually* composed

Zeigler, B.P., H. Praehofer, T.G. Kim, (2000), Theory of Modeling and Simulation, 2nd ed., Orlando, FL, USA: Academic Press.

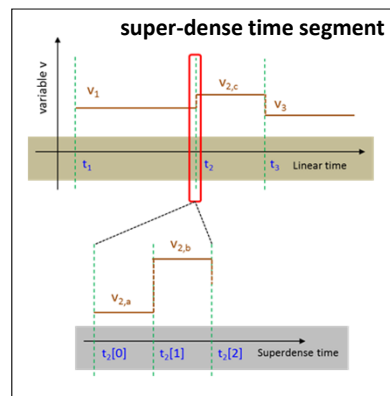
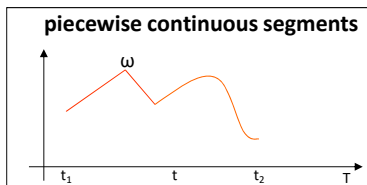
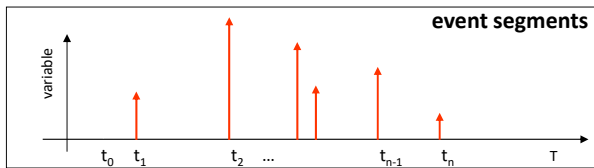
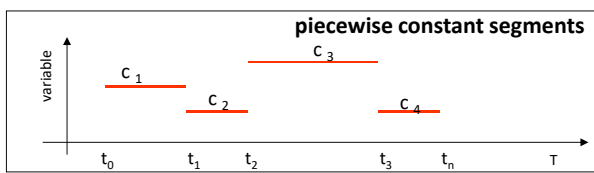
© Hessem S. Sarjoughian

12/6/2017

5

<https://acims.asu.edu>

event segments and trajectories



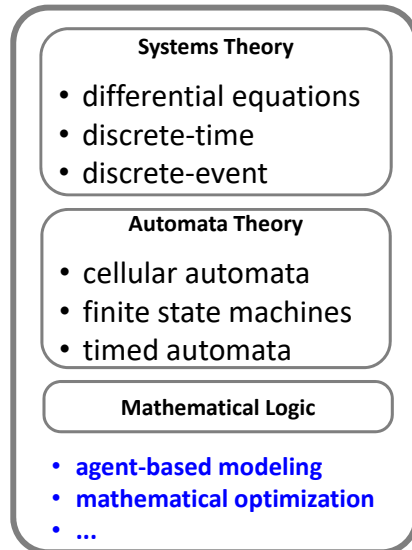
- Data is not restricted to numbers
- Converting one type of segment into another can introduce error
- Error in signals can accumulate and (in)directly transfer to others

© Hessem S. Sarjoughian

12/6/2017

6

modeling methods



Model abstractions should be computation and platform independent

Formal Models

- abstract modeling languages
- concrete modeling languages

Model "flexibility"

- scale
- complexity
- reuse

© Hessem S. Sarjoughian

12/6/2017

7

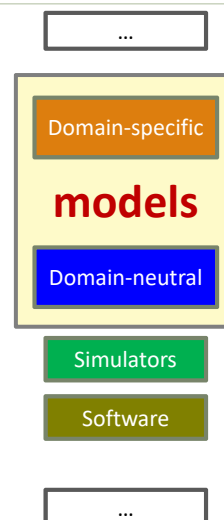
<https://acims.asu.edu>

modeling as *a layer* in simulations & model checking

→ keeping models separated and loosely assigned to simulators has unique advantages,

→ it is one facet of taming model complexity and scale

A prominent approach is Model-Driven Architecture (MDA)



© Hessem S. Sarjoughian

12/6/2017

8

hybrid systems / models

- Hybrid model of a system is composed from *multiple heterogeneous models*
 - A system has independent sub-systems
 - Each sub-system is described with a best-fit model type
- Model heterogeneity is fundamental to understanding and building systems of systems

Systems are inherently heterogeneous, but they are often represented as homogeneous models

<https://acims.asu.edu>

architectural complexity (1)

- Frequently, complexity takes the form of a *hierarchy, whereby a complex system is composed of interrelated subsystems* that have in turn their own subsystems, and so on, until some lowest level of elementary components is reached.
- The choice of *what components in a system are primitive is relatively arbitrary* and is largely up to the discretion of the observer of the system.
- Intra-component linkages are generally stronger than inter-component linkages. This fact has the effect of separating the *high-frequency dynamics* of the components - involving the internal structure of the components - from *low-frequency dynamics* - involving interaction among components.

architectural complexity (2)

- Hierarchic systems are usually composed of only a few different kinds of subsystems in various combinations and arrangements.
- A complex system that works is invariably found to have evolved from a simple system that worked A complex system designed from scratch never works and cannot be “patched up” to make it work. One has to start over, beginning with a simple working system.
- *Heterogeneity has a direct relationship with a systems level of complexity.*

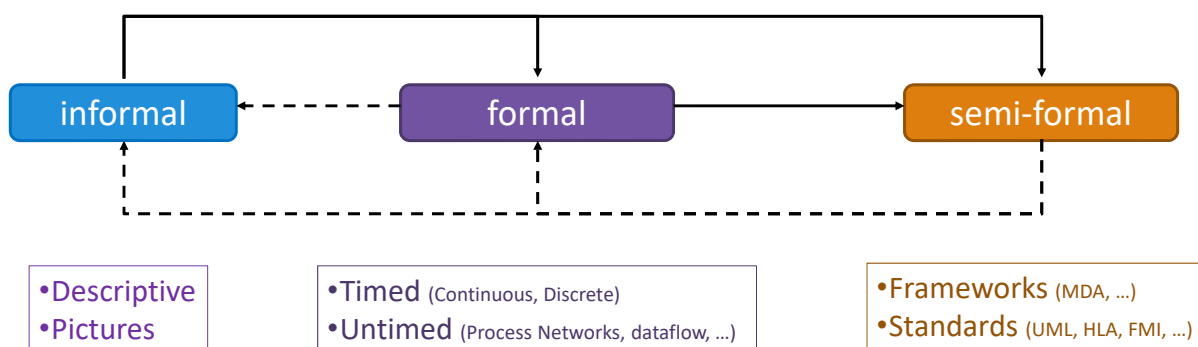
© Hessem S. Sarjoughian

12/6/2017

11

<https://acims.asu.edu>

model representations



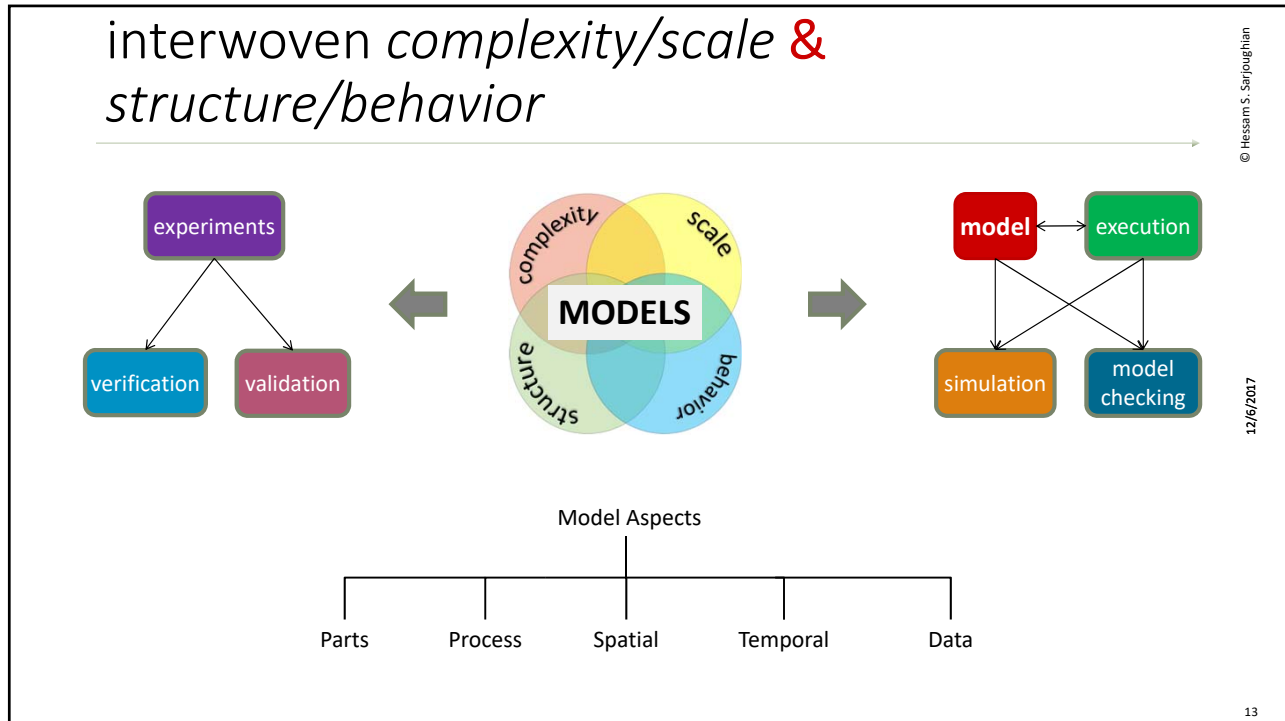
© Hessem S. Sarjoughian

12/6/2017

Scale: size of some system model or any parts thereof (~quantitative)

Complexity: presence or lack of order in a system model (~qualitative)

12



<https://acims.asu.edu>

multi-formalism modeling: supply-chain model

DEVS: Supply-Chain Enterprise

- Mimics supply-chain processes in terms of independent inventory and shipping dynamics and their synthesis using discrete event simulation

Heuristics: Inventory Strategy

- Generates inventory safety-stock targets using base stock algorithms to account for future bias in customer service levels.

KIB: Knowledge Interchange Broker

- Data aggregation/disaggregation with control and execution across disparate modeling formalisms.

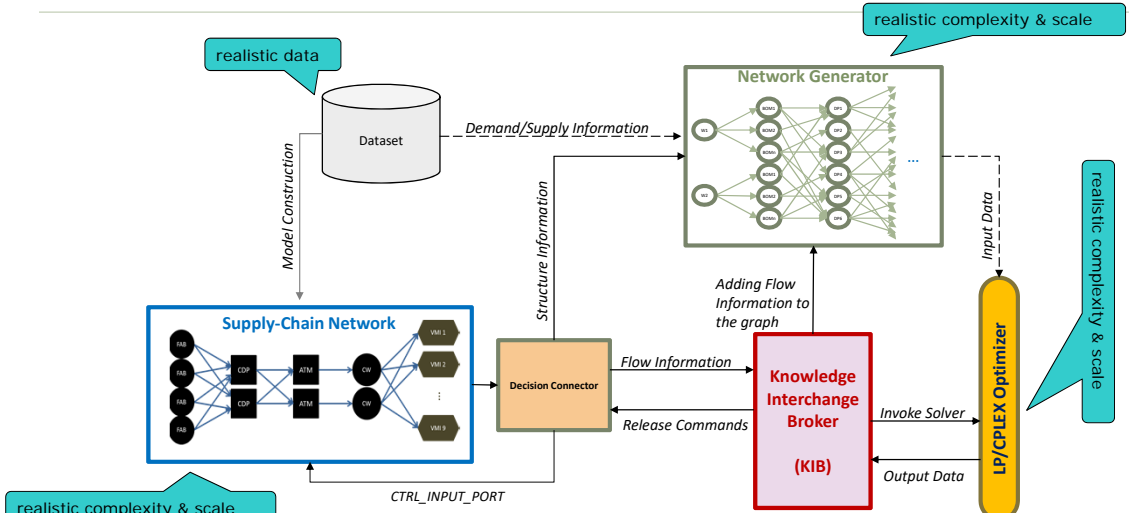
Optimizer: Control Strategy

- Computes release commands for inventories – i.e., shipment commands for products to be shipped to upstream inventories given safety stocks and customer forecast demands
- Optimization approach matches what Intel uses in production.

© Hessem S. Sarjoughian
12/6/2017

Huang, D., H.S. Sarjoughian, W. Wang, G. Godding, D. Rivera, K. Kempf, H. Mittelmann, (2009), "Simulation of Semiconductor Manufacturing Supply-Chain Systems with DEVS, MPC, and KIB", IEEE Transactions on Semiconductor Manufacturing, Vol. 22, No. 1, 165-174.

Intel's enterprise supply-chain system simulation & optimization architecture



Sarjoughian, H.S., J. Smith, G.W. Godding, M. Muqith, (2013), "Model Composability and Execution across Simulation, Optimization, and Forecast Models", SpringSim Multi-Conference, San Diego, CA.
 Gholami, S., H.S. Sarjoughian, G.W. Godding, D.R. Peters, V. Chang, (2014), "Developing Composed Simulation and Optimization Models Using Actual Supply-Demand Network Datasets", WSC, Savannah, GA.

© Hessem S. Sarjoughian

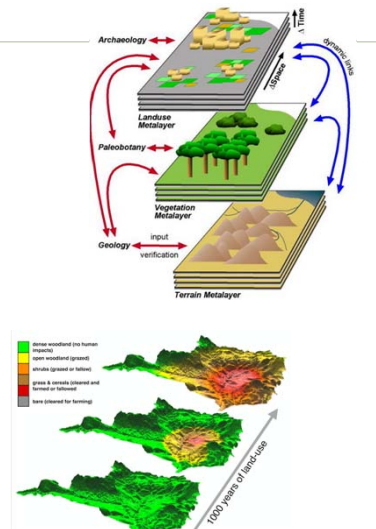
12/6/2017

17

<https://acims.asu.edu>

human/environment interactions

- Real-world application domain
 - Human farmers in Penaguila Valley, Spain
 - ~7000 BP, early Neolithic
 - ethnographical and anthropological data
 - Land surface and vegetation processes
- Models developed independently
 - Farmer households (agent)
 - Erosion/Deposition model (environment)
- Interactions
 - Households and villages farming landscape, landscape impact on household

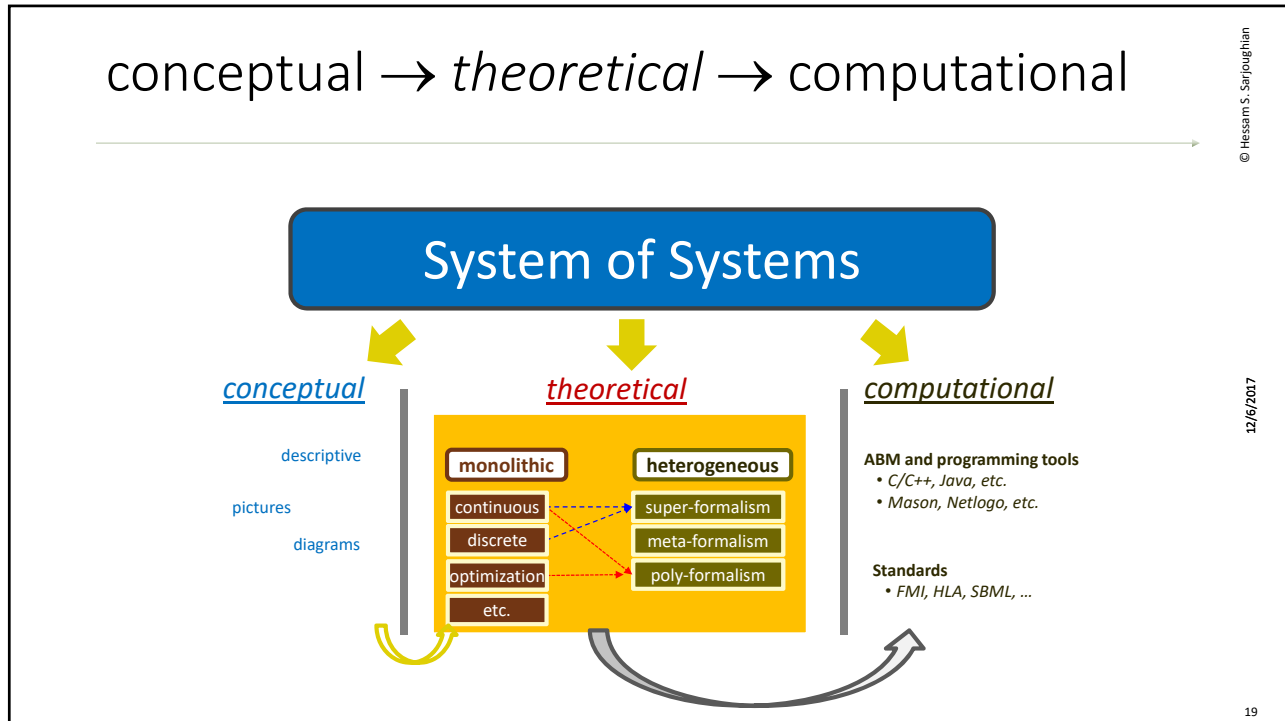


Mayer, G.R., H.S. Sarjoughian, (2009), "Composable Cellular Automata", Simulation Transactions, Vol. 85(11-12), 735-749.
 Sarjoughian, H.S., G.R. Mayer, I.I. Ullah, and C.M. Barton, (2015), Managing hybrid model composition complexity: Human-environment simulation models. In L. Yilmaz, editor, Concepts and Methodologies for Modeling and Simulation, pages 107-134. Springer.

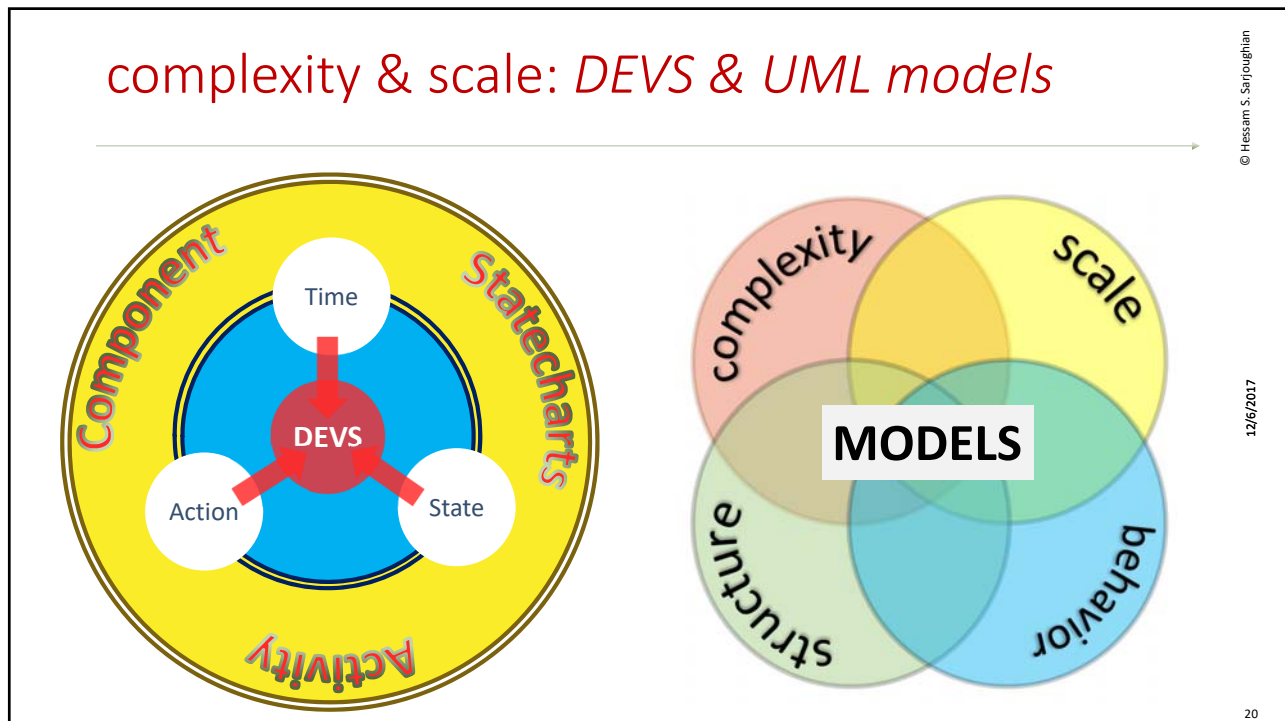
© Hessem S. Sarjoughian

12/6/2017

18

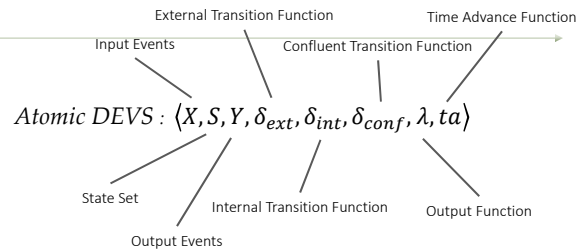


<https://acims.asu.edu>



DEVS M&S (1)

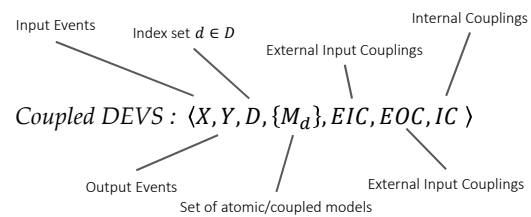
- Parallel DEVS models define basic structure and state-time behaviors



© Hessem S. Sarjoughian

12/6/2017

- Coupled DEVS models define composition of Atomic/Coupled models



21

<https://acims.asu.edu>

processor with queuing support (1)

$$\text{Processor}_{queue} = (X^b, S, Y^b, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta) \quad \text{where}$$

$$X^b = \{(p, v) | p \in IPorts, v \in X_{in}\}$$

$IPorts = \{\text{"in"}\}$ is an arbitrary input port name

$X_{in} = V_X$ is an arbitrary set of input variable names and values

$$S = \{\text{"passive"}, \text{"busy"}\} \times \mathfrak{R}_{>0}^{+\infty} \times \{\bar{q}, q', q''\}$$

$$Y^b = \{(p, v) | p \in OPorts, v \in Y_{out}\}$$

$OPorts = \{\text{"out"}\}$ is an arbitrary output port name

$Y_{out} = V_Y$ is an arbitrary set of out variable names and values

© Hessem S. Sarjoughian

12/6/2017

22

processor with queuing support (2)

$$\begin{aligned} \delta_{ext}((phase, \sigma, \bar{q}), e, ((\text{"in"}, x_1), \dots, (\text{"in"}, x_n))), x_i \in X_{in} \\ = (\text{"passive"}, processingTime, q' | q'') \quad \text{if phase is "passive" and queue is empty} \\ \delta_{ext}((phase, \sigma, q' | q''), e, ((\text{"in"}, x_1), \dots, (\text{"in"}, x_n))), x_i \in X_{in} \\ = (\text{"busy"}, \sigma - e, q'') \quad \text{if phase is "busy" and queue is not empty} \end{aligned}$$

$$\begin{aligned} \delta_{int}(phase, \sigma, q') \\ = (\text{"passive"}, \infty, \bar{q}) \quad \text{if phase is "busy" and queue has one input} \\ \delta_{int}(phase, \sigma, q'') \\ = (\text{"busy"}, processingTime, q' | q'') \quad \text{if phase is "busy" and queue more than one input} \end{aligned}$$

$$\delta_{con}((s, ta(s)), ((\text{"in"}, x_1), \dots, (\text{"in"}, x_n))), x_i \in X_{in} = \delta_{ext}(\delta_{int}(s), 0, ((\text{"in"}, x_1), \dots, (\text{"in"}, x_n)))$$

$$\lambda(phase, \sigma, q | q') = (\text{"out"}, y_j), y_j \in Y_{out}, \quad \text{if phase} = \text{"busy"}$$

$$\begin{aligned} ta(phase, \sigma, q') &= \infty \quad \text{if queue has one input} \\ ta(phase, \sigma, q'') &= \mathfrak{R}_{>0}^{\leq \infty} \quad \text{if queue has more than one input} \end{aligned}$$

© Hessem S. Sarjoughian

12/6/2017

23

<https://acims.asu.edu>

structural & behavioral models

| Type | Structure Specifications | Behavior Specifications |
|--------------------------|--------------------------|--------------------------------|
| | Model [Framework] | Model [Framework] |
| Individual Models | Class [DEVS-Suite] | Statecharts, Activity [CoSMoS] |
| Family of Models | Component [DEVS-Suite] | Statecharts, Activity [CoSMoS] |

DEVS-Suite: simulation and model checking engines

- Java language
- BIRT plugin
- Java FX

CoSMoS: modeling and experimentation engines

- Java language
- Relational database (MS Access)
- Eclipse Papryus
- Eclipse GMF

© Hessem S. Sarjoughian

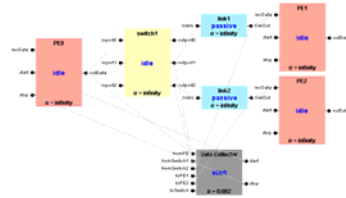
12/6/2017

24

DEVS-Suite: simulation and model checking

• DEVS Modeling

- Features
 - Continuous time, discrete event, parallel
 - Constrained DEVS
 - Action-level
 - Real-time



• DEVS Simulation & Model Checking

- Can be conducted in
 - Logical time: time is advanced to the most urgent event
 - Real-time: simulation time is synchronized with the physical clock
- Various implementations with different capabilities
 - DEVS-Suite, eCD++, Mod4Sim



DEVS-Suite, (2017), available at <https://acims.asu.edu/software/devs-suite/>.

© Hessem S. Sarjoughian

12/6/2017

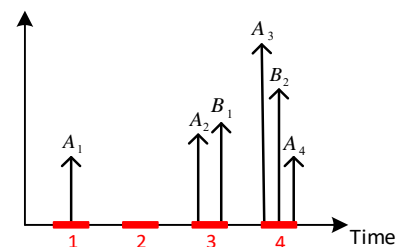
25

<https://acims.asu.edu>

models of time (cont.)

• Superdense time

- Useful for modeling simultaneity of events
- It is defined in terms of time instances and indexes
 - Time instances increase monotonically
 - A finite number of monotonically increasing index numbers can be assigned to each time instance
 - For any two events assigned (t, n_1) and (t, n_2) , the events happen at the same time
 - Weak simultaneity: $n_1 \neq n_2$
 - Strong simultaneity: $n_1 = n_2$
- Two forms of causally related events
 - Weak simultaneous events
 - Strong simultaneous events
- Superdense time is represented lexicographically
 - $(t_1, n_1) < (t_2, n_2)$
 - $t_1 < t_2$ or
 - $t_1 = t_2 \wedge n_1 < n_2$

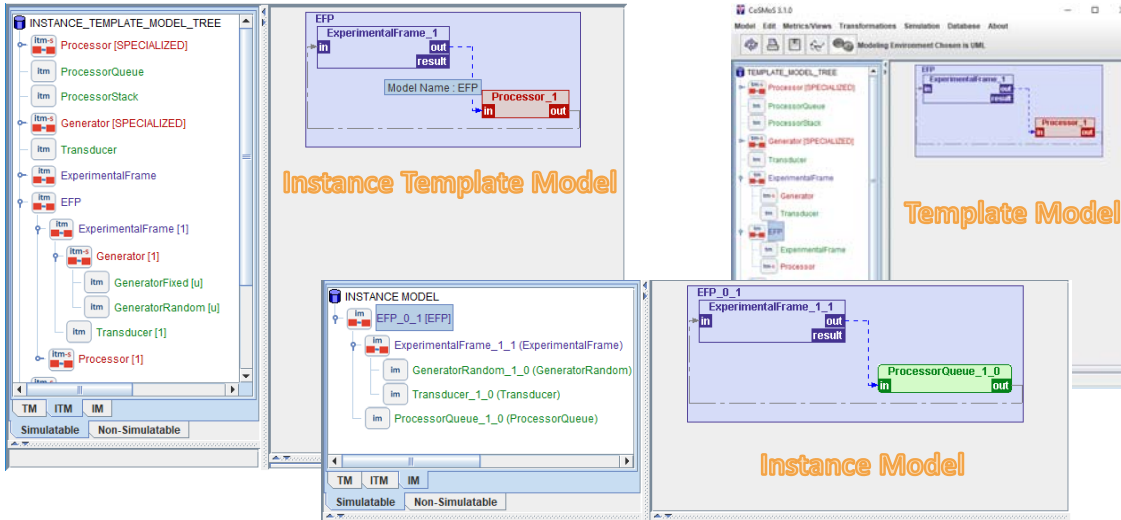


© Hessem S. Sarjoughian

12/6/2017

26

CoMoS: modeling engine with database repository



CoSMoS, (2017), available at <https://acims.asu.edu/software/cosmos/>

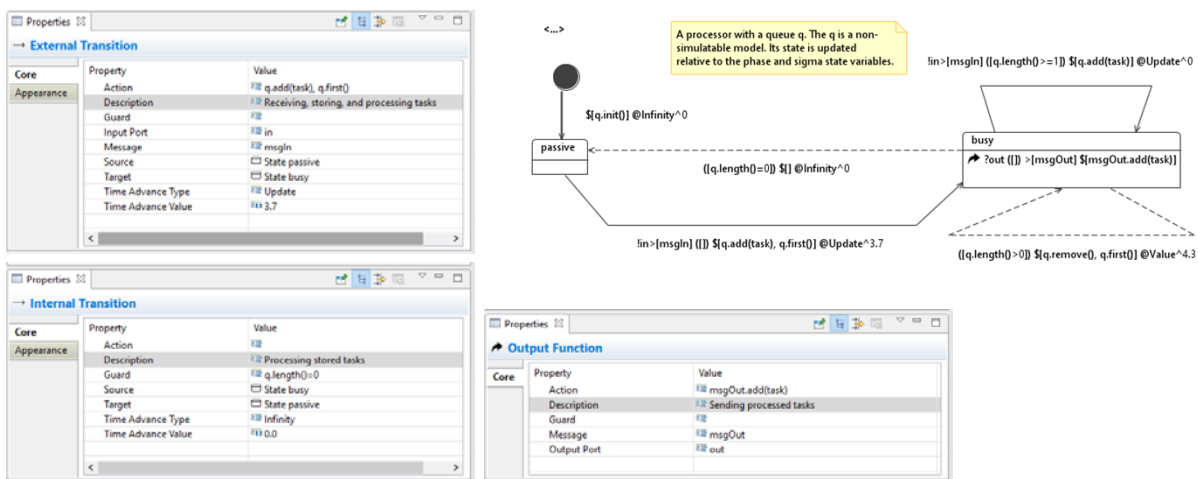
© Hessem S. Sarjoughian

12/6/2017

27

<https://acims.asu.edu>

DEVS statecharts modeling: processor with queue (1)



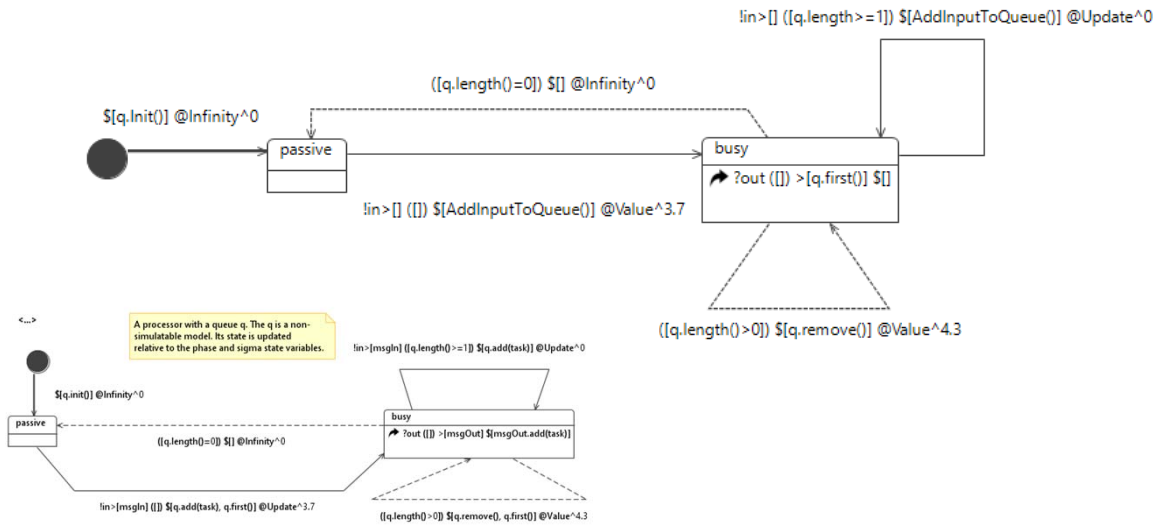
Fard, M. and H.S. Sarjoughian, (2015), "Visual and Persistence Behavior Modeling for DEVS", TMS/DEVS Symposium, Wash. DC.

© Hessem S. Sarjoughian

12/6/2017

28

DEVS statecharts modeling: processor with queue (2)



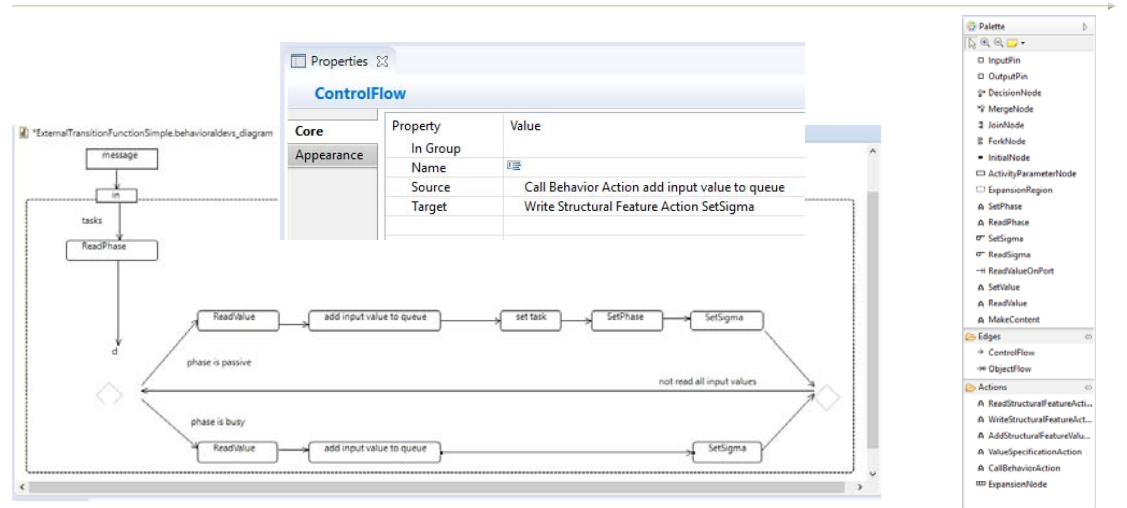
© Hessem S. Sarjoughian

12/6/2017

29

<https://acims.asu.edu>

DEVS activity modeling



Alshareef, A., H.S. Sarjoughian, and B. Zarrin, (2016), "An Approach for Activity-Based DEVS Model Specification". Spring Simulation Multi-Conference, Symposium on Model-Driven Approaches for Simulation Engineering, 1–8.

© Hessem S. Sarjoughian

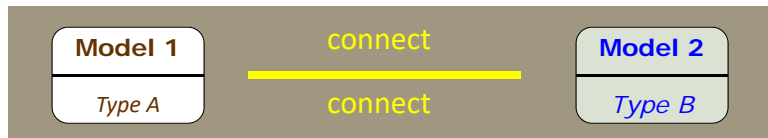
12/6/2017

30

interconnecting “heterogeneous models”

- Coupling **modeling methods**

- *relationships are defined in a modeling method*
- *Interactions between models defined in different modeling methods are brokered*



Interaction Model Language:

- Syntax: Structure + Behavior
- Semantics: Execution Protocol



- ✓ *data transformation*
- ✓ *timing*
- ✓ *control scheme*
- ✓ *execution modes*

<https://acims.asu.edu>

composite modeling

- **input/output connectors**

- *increase model complexity* ↑
- *reduce model scalability* ↓

- **interaction models**

- *reduce model complexity* ↓
- *increase model scalability* ↑

heterogeneous composite modeling (2)

single model type

- system and its parts have the same structural and behavioral characteristics

monolithic composition

- composite model is modeled in the same way the parts are modeled

multiple model types

- system and its parts do not have the same structural and behavioral characteristics

polymorphic composition

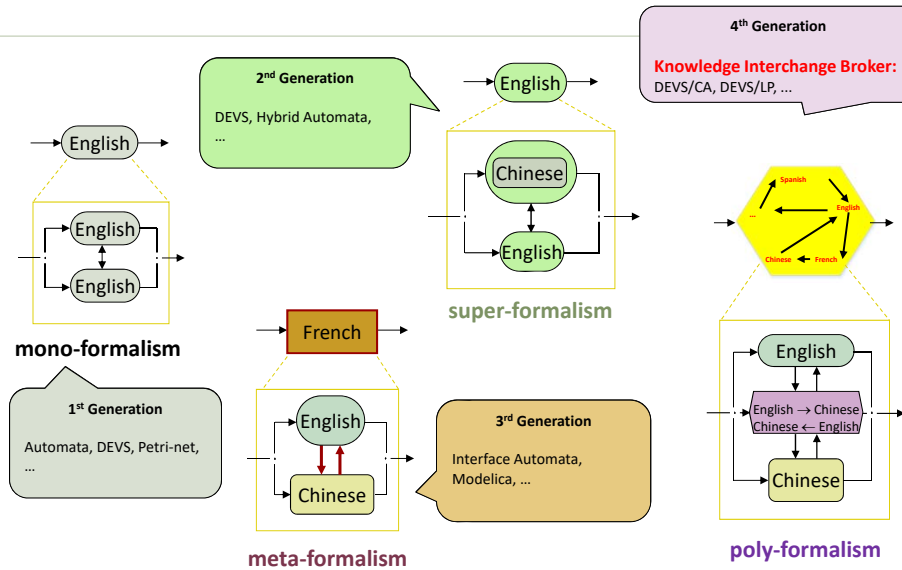
- composite model cannot be modeled in the same way the parts are modeled

<https://acims.asu.edu>

model composition categories

- *Semantically* composing different models of a system with multiple types of dynamics
- Modeling Composition approaches
 - **Mono Modeling**
 - A single formalism is used
 - **Super-Formalism Modeling**
 - Two or more distinct model types are encapsulated within a modeling formalism
 - **Meta-Formalism Modeling**
 - Different types of model specifications are transformed or abstracted to another higher level modeling formalism
 - **Poly-Formalism Modeling**
 - The interaction between disparate modeling formalisms is explicitly modeled
- Software Component Model Specification and Interaction
 - UML, MDA, XML ...

illustration of model composition theories



Sarjoughian, H.S., (2006), "Model Composability", Winter Simulation Conference, Monterey, CA, USA

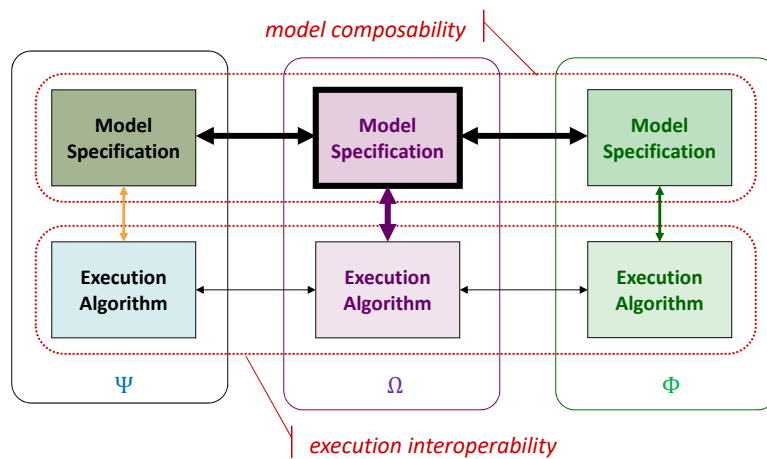
© Hessem S. Sarjoughian

12/6/2017

35

<https://acims.asu.edu>

separation of model composability and simulation interoperability

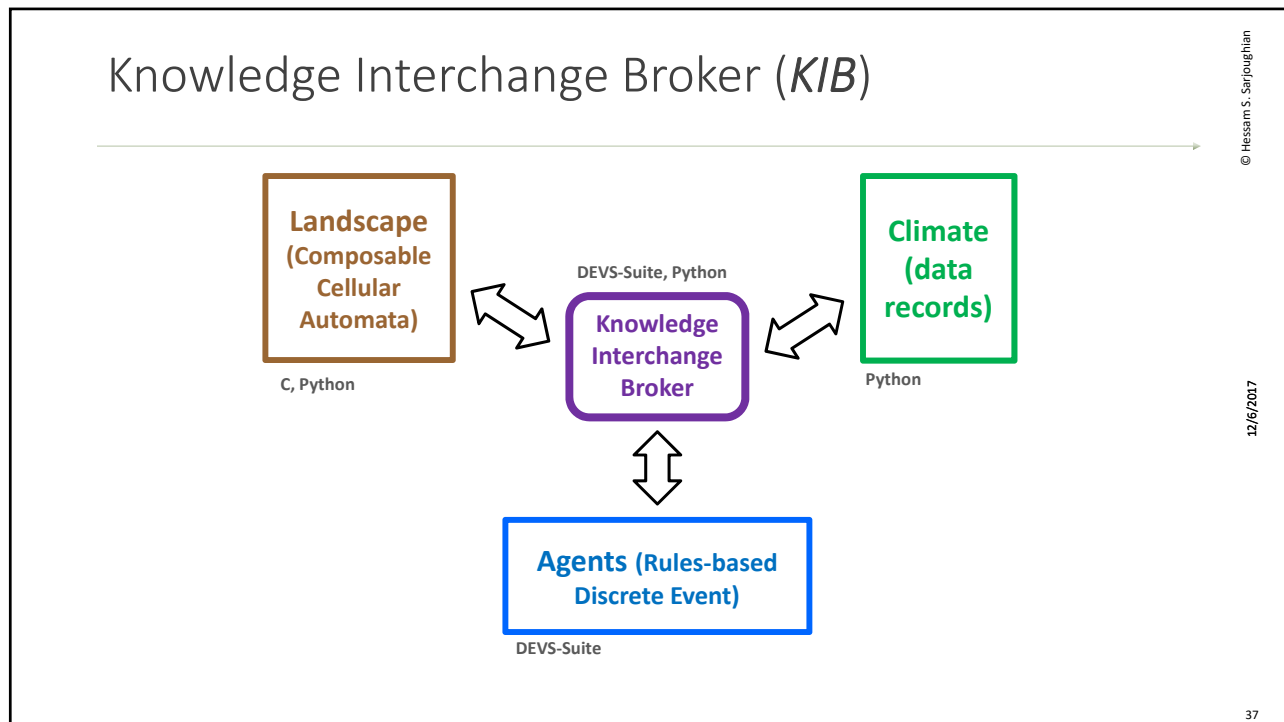


formalism Ω is composition of formalisms Ψ and Φ

© Hessem S. Sarjoughian

12/6/2017

36



<https://acims.asu.edu>

models disparities

- Timing
 - DEVS-Suite: innate sense of time
 - GRASS: no concept of time within models
- Simulation synchronization
 - DEVS-Suite: conforms to DEVS simulator protocol (concurrent)
 - GRASS: no overarching simulator/executor
 - module execution managed by scripts (sequential)
- Agent activity
 - Agent: require some amount of resource (food)
 - Landscape and vegetation: just need to know where resource was harvested

© Hessem S. Sarjoughian
12/6/2017
38

multi-formalism model composition

Knowledge Interchange Broker (KIB)

defines dynamic interactions for composed modeling formalisms

- **Specification**
 - data transformations
 - synchronization
 - concurrency
 - timing
- **Protocol**
 - execution coordinated with the executions of the composed models

© Hessem S. Sarjoughian

12/6/2017

39

<https://acims.asu.edu>

configurable I/O and state transformation specifications

- Capture the specialties of each participating modeling method
 - I/O characteristics (communication, data type)
 - Single/multi-value to single/multi-value types
- Data transformation (functions and relationships)
 - Knowledge reduction *or* knowledge augmentation
- Timing for I/O and state transformations
 - Consistent time specifications and resolutions (step-time, event-time mappings)

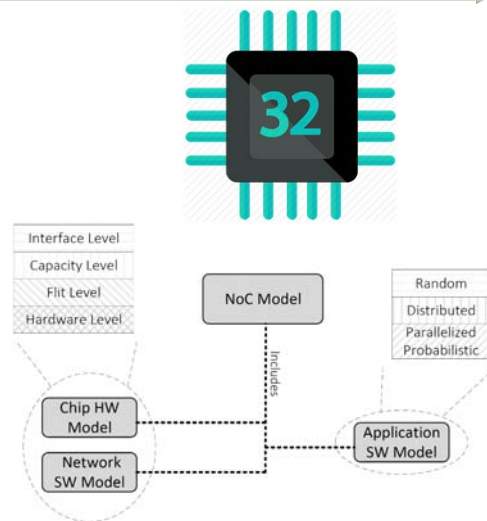
© Hessem S. Sarjoughian

12/6/2017

40

Network-on-Chip (NoC) model

- NoC supports communication in SoC systems
- Major parts
 - Chip Hardware: electronic components of the circuit
 - Network Software: modules controlling the circuit
 - Application Software: user application



Gholami, S. and H.S. Sarjoughian, (2017), "Action-level, Real-time Network-on-Chip Modeling", Simulation Modelling Practice and Theory, pages 272-291, Vol. 77.

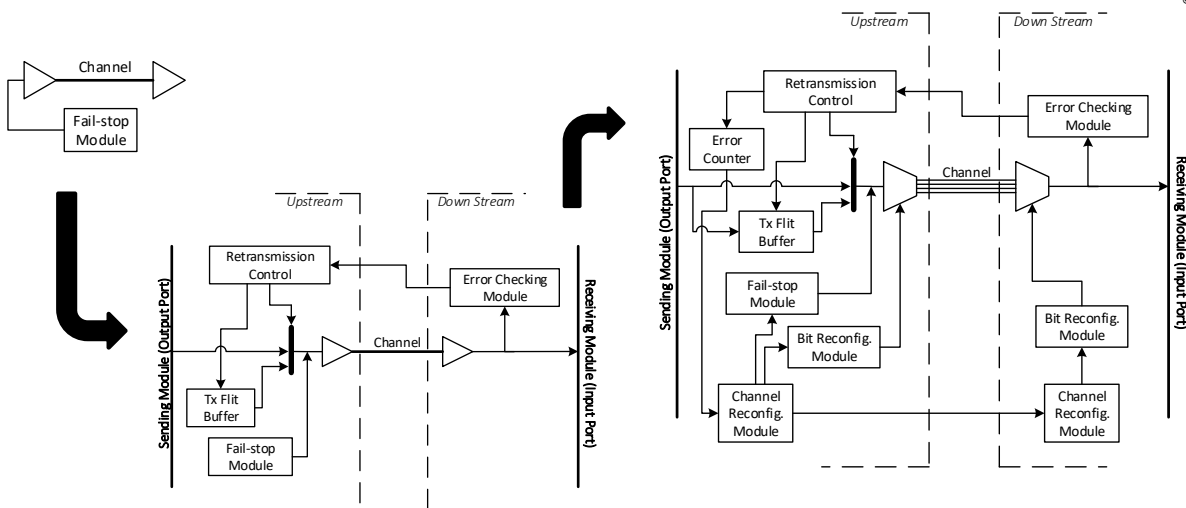
© Hessem S. Sarjoughian

12/6/2017

41

<https://acims.asu.edu>

Link: complexity & scale traits

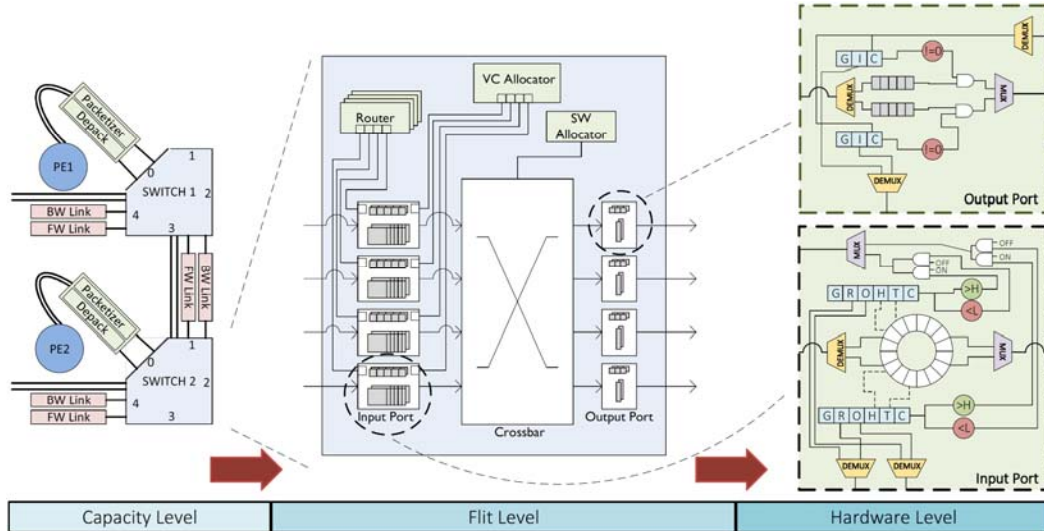


© Hessem S. Sarjoughian

12/6/2017

42

switch model: complexity & scale traits



© Hossam S. Sarjoughian

12/6/2017

43

<https://acims.asu.edu>

NoC models at varying levels of complexity & scale

- The difference between NoC hardware (and network SW) abstraction levels can be expressed in terms of complexity & scale traits

| Accuracy/ Precision → Resolutions ↓ | NoC HW Abstraction Levels | | | |
|--|---------------------------|----------|------|----------|
| | Interface | Capacity | Flit | Hardware |
| Object | ✓ | ✓ | ✓ | ✓ |
| Temporal | | ✓ | ✓ | ✓ |
| Process | | | ✓ | ✓ |
| Data | | | ✓ | ✓ |
| Spatial | | | | ✓ |

- Similarly, one can describe application software abstraction levels complexity & scale traits

| Accuracy/ Precision → Resolutions ↓ | NoC SW Abstraction Levels | | |
|--|---------------------------|-------------|---------------|
| | Random | Distributed | Probabilistic |
| Object | ✓ | ✓ | ✓ |
| Temporal | | ✓ | ✓ |
| Process | | | ✓ |
| Data | | | ✓ |

© Hossam S. Sarjoughian

12/6/2017

44

heterogeneous multiscale modeling

Modeling principle: system decompositions (sub-systems compositions) within and across *temporal* and *spatial* scales

| | Partial Diff. Eq. + Cellular Automata | Ordinary Diff. Eq. | Discrete Eq. + Cellular Automata |
|----------------|---|---|---|
| scale | molecular | cellular | tissue |
| time | tiny → small (10 ⁻⁹ s ~ 10 ⁻⁵ s) | small → medium (10 ⁻¹ s ~ 10 ³ s) | medium → large (10s ~ 10 ⁶ s) |
| spatial | tiny → small (10 ⁻⁹ m ~ 10 ⁻⁵ m) | small → medium (10 ⁻⁹ m ~ 10 ⁻⁵ m) | medium → large (10 ⁻⁵ m ~ 10 ⁻³ m) |
| Purpose | gradient formation, diffusion, ... | receptor-ligand binding, intercellular signaling, ... | cell activation, cell migration, ... |

Cilfone, N. D., Kirschner, J. Linderman, (2015). "Strategies for Efficient Numerical Implementation of Hybrid Multi-scale Agent-Based Models to Describe Biological Systems Cellular and Molecular Bioengineering, 8(1):119-136.

© Hessem S. Sarjoughian

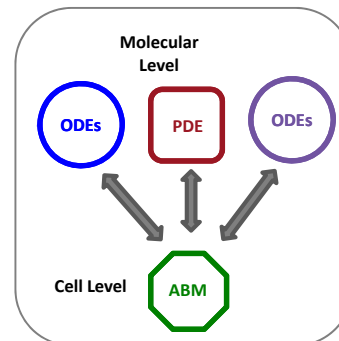
12/6/2017

45

<https://acims.asu.edu>

system biology models

| Model | Agent-Based Modeling (ABM) | Ordinary Diff. Eq. (ODE) | Partial Diff. Eq. (PDE) |
|--|----------------------------|--------------------------|-------------------------|
| Chemotaxis of CXCR+ Cells | ✓ hours | | |
| Chemokine secretion and receptor-ligand dynamics | | ✓ a few seconds | |
| Chemokine internalization internal signaling | | ✓ 10s of seconds | |
| Changes in local chemokine gradient | | | ✓ 10s of seconds |

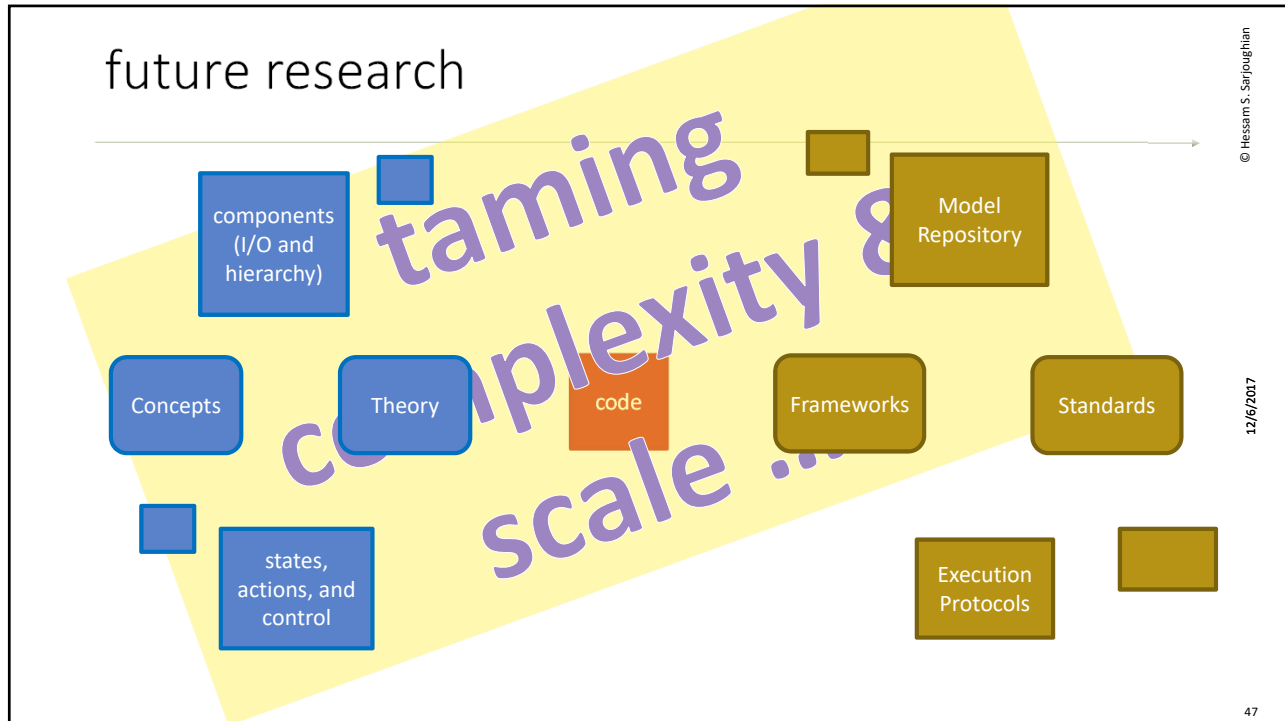


Chang, S. L., Cavnar, S. P., Takayama, S., Luker, G. D., & Linderman, J. J., (2015). "Cell, isoform, and environment factors shape gradients and modulate chemotaxis." PloS one.

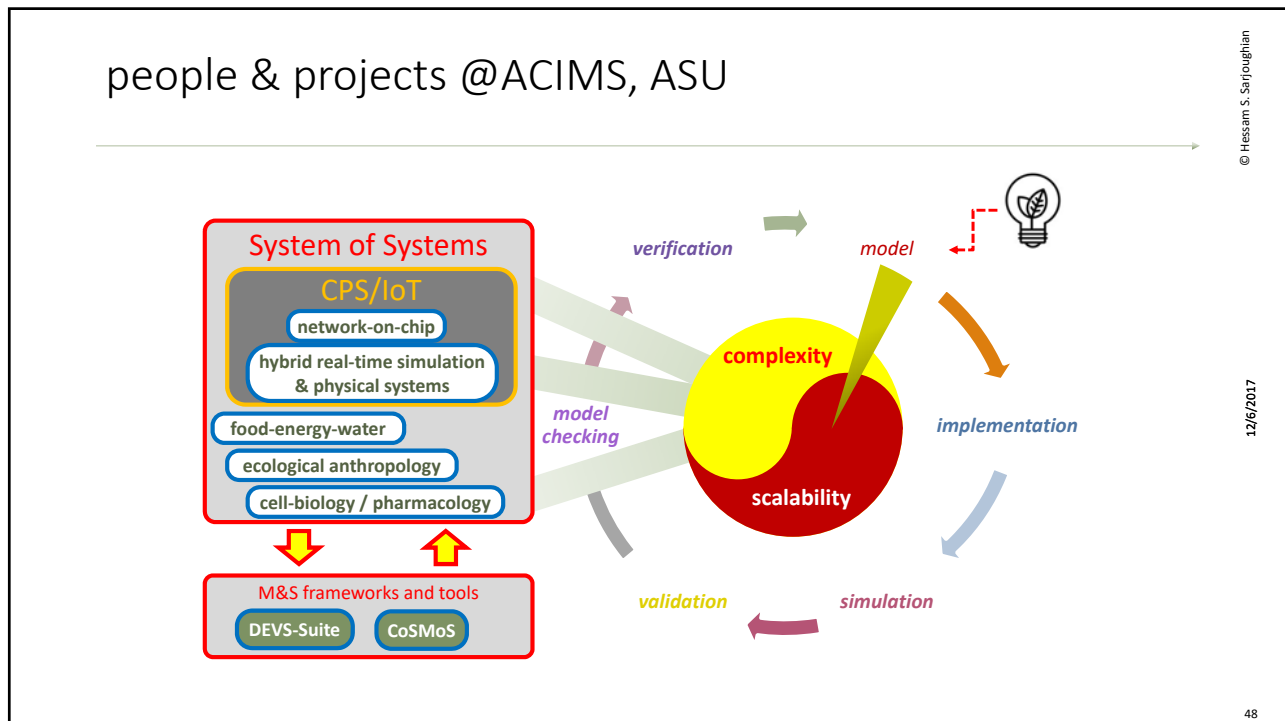
© Hessem S. Sarjoughian

12/6/2017

46



<https://acims.asu.edu>



Acknowledgments

NSF DEB-1313727
NSF CNS-1639227



visiting scholars and collaborators

students

Abdul
Alshareef



Chao
Zhang



Elena
Wegryn



Gregg
Scherer



Mostafa
Fard



Masudul
Quraishi



Soroosh
Gholami



William
Boyd



© Hessem S. Sarjoughian

12/6/2017

49

<https://acims.asu.edu>

Disparities – simulators

-
- DEVS-Suite Simulator
 - Java & Java FX languages, BIRT; built to Parallel DEVS formalism
 - communication via message objects; hierarchical path
 - GRASS (Geographic Resources Analysis Support System)
 - C programming language; Geographic Information System implementation
 - modules execute and terminate
 - each state variable stored in a separate map
 - map algebra
 - Geospatial relationship; data relations defined via scripts
 - model: set of maps and specified dynamics

© Hessem S. Sarjoughian

12/6/2017

50