Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

EXPLOITING THE LEVELS OF SYSTEM SPECIFICATION FOR MODELING OF MIND

Bernard P. Zeigler

Electrical and Computer Engineering Department
University of Arizona
1200 E University Blvd
Tucson, AZ 85721, USA

ABSTRACT

Our objective is to show how the hierarchy of system specifications and morphisms affords a framework that supports modeling and simulation of mind/brain in a coherent manner. Such a framework provides a credible path for simulation/implementation of cognitive behavior at the level of neurons and neuronal compositions. Here we characterize this methodology and use it to extend the set of DEVS building blocks and architectural patterns within complex decision trees and advanced classifier implementations as DEVS coupled models. Such a methodology offers insight into the way real brains realize cognitive behaviors and can improve upon current hardware architecture to support simulation/implementation of neuromorphic designs that comply with physiologically based brain network properties and constraints.

1 INTRODUCTION

With the increased availability of brain experimental data at increasingly granular levels and the growing computational technologies to process them, the challenge persists to connect such observations to behavior at the cognitive level. A framework is needed that supports modeling and simulation (M&S) in a consistent manner that links knowledge gained at all levels of structure and behavior and enables knowledge gained at any level to contribute to relevant knowledge residing at levels above or below. This entails a computational bridge (Carandini 2012) between models at the level of neural circuits and those at the level of cognitive behavior that supports experimental investigation of cognitive behaviors and modeling them in abstraction stages by biologically plausible networks in brain structures (Petersen and Sporns 2015). Such a framework is a family of models and pair-wise mappings that provide a credible path for neural-based mechanistic generation of cognitive behavior. The bridge is bi-directional in that constraints established on behaviors at higher levels induce mapped constraints at the lowest level and vice versa.

Currently there is no generic bridge between neural and cognitive levels – there are only some examples of such a computational mapping in specific contexts and there are very few attempts to associate well-known or newly discovered cognitive behaviors with neural level structures on a causal foundation. For example, monkeys' reaction times in a forced-choice motion task (O'Connell et al. 2018) were experimentally observed to be correlated with increased spike activity of lateral intraparietal cortical neurons. Here a succession of models (Wong and Wang 2006) representing valid simplifications to a final reduced two-variable neural model was shown to offer a simple and biophysically plausible mechanistic explanation for the observed reaction time characteristics. However, currently this is a rare instance in which a full model chain was created between structure and behavior. Moreover, there is no easy generalization to other examples. Our ultimate goal is to develop a methodology to create such computational bridges to help suggest experiments that causally link the activity of local neuronal circuits with observations at cognitive behavior levels.

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

A spinoff of the bridging methodology is implementation of formally specified cognitive behavior in biologically inspired (Feinerman, Rotem and Moses, 2008). and other hardware (Adegbija 2022). This can be accomplished in a manner similar to the realization theory for implementing finite state machines in logic nets built with hardware primitives. by the algebraic structure theory of sequential machines and Krohn-Rhodes decomposition theory. This realization theory is based on systems modeling and simulation theory (Stearns and Hartmanis 1966, Krohn and Rhodes 1965) as expressed in the Discrete Event System Specification (DEVS) formalism (Zeigler, Muzy, and Kofman 2018).

2 BACKGROUND

DEVS has been used to model cognitive behavior and has been used to build cognitive architectures (Douglass and Mittal 2013). Zeigler (2021) presented concepts for building blocks and architecture coupling patterns drawn from the cognitive science literature, particularly, the fast, frugal and accurate paradigm (Gigerenzer and Goldstein, 2000) for applications intelligent cyberphysical system design. The proposed concepts are formulated with the discrete event abstraction embodied in the DEVS formalism and its extensions to hybrid modeling and simulation (Zeigler 2020) that are increasingly being adopted as the preferred approach to hybrid (continuous and discrete) cyberphysical system design (Castro 2019). Building blocks and architectural patterns that can be replicated and reused in system development are intended to accelerate this adoption. In this paper, we place this methodology within the broader framework of the levels of system specification and confirm its utility by using it to identify additional primitives and architectural coupling patterns that help to understand and design intelligent cyberphysical systems. With this capability established, we go on to compare this approach with those of other prominent theories of mind.

Our methodology can be described along the following dimensions:

- **Use of system specification hierarchy and associated morphisms.** The levels of system specification range from lowest level behavior specification to highest level structural specification (Zeigler, Muzy and Kofman 2018; Mittal and Zeigler 2014). Corresponding to each system specification level is a morphic relation appropriate to a pair of systems specified at that level. Morphisms at each level are defined such that a morphism which preserves the structural features of one system in another system at one level also preserves its features at all lower levels. The morphisms at the I/O Function and State Transition levels are the ones that underlie the minimal realization and homomorphic image concepts supporting the quest for minimal forms mentioned earlier. We provide an example of the application of the minimal realization and levels of system specification in Section 3.2.
- **Development Method**. We seek to develop models and potential primitive components and coupling patterns as illustrated in Zeigler (2021). To do this, we try to define cognitive behaviors as I/O System Functions at the lower levels of the system specification hierarchy. We then seek minimal realizations and computationally feasible implementations. Finally, we try to validate the latter via proofs and DEVS simulations. Our choice of behaviors is motivated by the desire to come up with, and define, building blocks and architectural coupling patterns for ubiquitous, composable, and reusable application.
- **Minimal forms**. In line with the hoary dictum of philosophy, Occam's razor, we seek explanations of behavior that contain only those assumptions that are necessary to the explanation. However, the minimal realizations that we seek are based on concepts formulated in mathematical systems theory derived from both linear systems theory and finite automata theory (Zeigler 2021). Proving that a realization of a behavior is minimal in this sense implies that it is a homomorphic image in relation to any implementation of the same behavior. Moreover, definitions for state-based realization of cognitive behaviors based on mathematical system theory and DEVS fundamentally include temporal and probabilistic characteristics of neuron system inputs, state, and outputs (Zeigler, Muzy and Kofman 2018). Moreover, they provide a solid system-theoretical foundation and

simulation modeling framework for the high-performance computational support of intelligent cyberphysical systems.

- **Network construction**. The hierarchy of system specifications includes levels for definition of networks of components with coupling specification. This is exemplified by the DEVS coupled model definition with its well defined coupling specification. The proof of closure under coupling shows how resultant networks are equivalent to basic models, and can be treated as such in hierarchical construction.
- **Model formalism for Simulation and Design.** DEVS enables formal and complete description of hybrid model components and subsystems. DEVS-based software tool sets provide atomic model and hierarchical coupled model specifications that support graphical description of the internals and interfaces of component behavior combining energy, material, and information flows. The hybrid DEV&DESS (Zeigler, Muzy and Kofman 2018) formalisms enable expressing differential and algebraic equations for energy-related internal variables intermixed with discrete behavior described in state-based system form. Finally, transparent implementation of the canonical DEVS abstract simulator for handling events and equations enable design of dedicated simulation functionality.

These dimensions were selected after examining the alternative theories to be compared with the proposed levels of systems specification approach. They are intended to help identify the most important ways in which the theories differ and discern the consequences for arriving at the "best" understanding possible.

3 DEVS DECISION BEHAVIOR USE CASE

We now apply the methodology to extend the set of DEVS building blocks and architectural patterns based on minimal realizations and the levels of system specification developed earlier (Zeigler 2021). Our approach to establishing that a model is a minimal realization of a given cognitive behavior is to start by specifying the behavior as a system at the I/O behavior level. This requires us to:

1. Define the behavior formally as a function mapping input segments to output segments
2. Seek a DEVS model at the state description level that generates the defined behavior. This amounts to associating the state description to the original i/o system description.
3. Try to show that the state description is a minimal realization of the behavior: the state level description sets up the design of a network equivalent which amounts to a specification at the coupled component level of systems specification
4. Try to prove that the network is homomorphic to the established minimal realization. Alternatively, validate that it generates the required behavior by direct simulation.

The network form is typically closer to potential plausible biological implementations and the established minimality of realization allows to claim that the latter realizations must be a homomorphic image of any such realization. Please refer to Chapter 16 of (Zeigler, Muzy and Kofman 2018) for a detailed exposition of system realization theory.

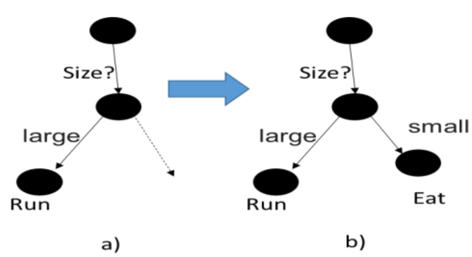

Figure 1: A primitive miniature-brained organism which can sense the size of a large oncoming object in a) and in later evolution, can distinguish between both large and small objects (b)).

DEVS abstractions attempt to capture many features of biological neurons that are not represented in conventional artificial neural networks and try to exploit these capabilities to perform intelligent control tasks (Vahie 2000). We demonstrated how a “one-spike per neuron” architecture employs a temporal, rather than firing rate, code for propagating information through neural processing layers. Here a winner-take-all decision is made by the arrival of the first spike at an output actuator – subsequent later arrivals are blocked from affecting the output. In this paper, we apply the same approach to slightly more complex spikes which carry magnitude information and support choices based on such magnitudes. We can imagine a primitive miniature-brained organism which can sense the size of an oncoming object. Then, as in Figure 1a it is able to react to a large enough size by running away but lacks the ability to respond otherwise.
In later evolution, such an organism might be able to distinguish between both large and small objects, considering the small instance as something to eat (Figure 1b). In other words, the single output event (respond/non-respond) is a more primitive form of decision making than the common binary choice with two distinct alternative output events. These elementary decision forms are depicted as I/O functions in Figure 2. In the first case (a), an arrival of a spike with value *x*, exceeding a threshold, *Th* causes an output, *Yes*, at some time later, while the arrival of sub-threshold spike results only in a null event. In contrast, Figure 2b models a binary decision element with the arrival of a sub-threshold spike causing an output of *No* some time later. Figures 2c and 2d sketch minimal DEVS atomic models that realize the I/O functions depicted in 2a and 2b, respectively. We omit the proof of realization and minimality in these examples as they follow the pattern discussed by Zeigler (2021).

Notice that the primitive *Yes*/null behavior and its state system realization models the functionality of a spiking neuron (Gautrais and Thorpe 1998). Such neurons generate a spike when their membrane potential reaches the threshold but are incapable of generating different outputs for individual sub-threshold inputs.

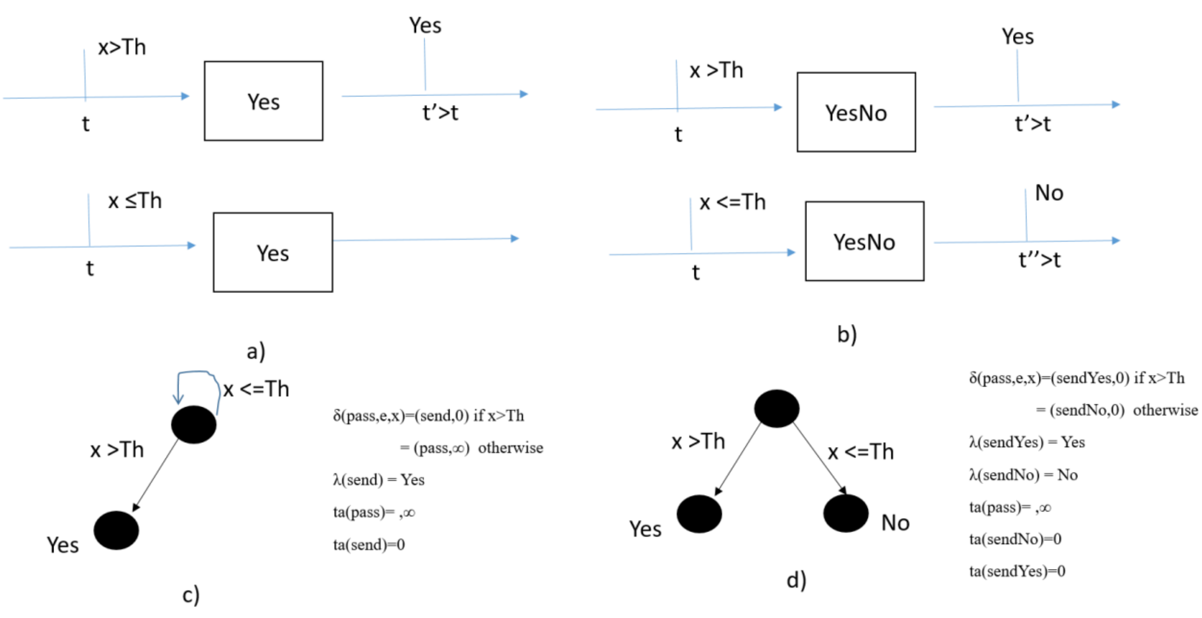

Figure 2: Elementary decision forms depicted as I/O functions and state minimal realizations.

Appendix A discusses a DEVS coupled model with two components to realize the binary choice Yes/No.

The model shows how to extend the Yes/null behavior associated with a single neuron to obtain the needed No output functionality. Given the limitation of single neurons, we infer that evolution needed to move in the direction of either constructing more complex neurons or enabling neurons to collaborate to implement the *Yes/No* binary decision behavior. Such a step would also lead to reuse of the model as a building block in the form an iterative decision tree illustrated in Figure 3a.

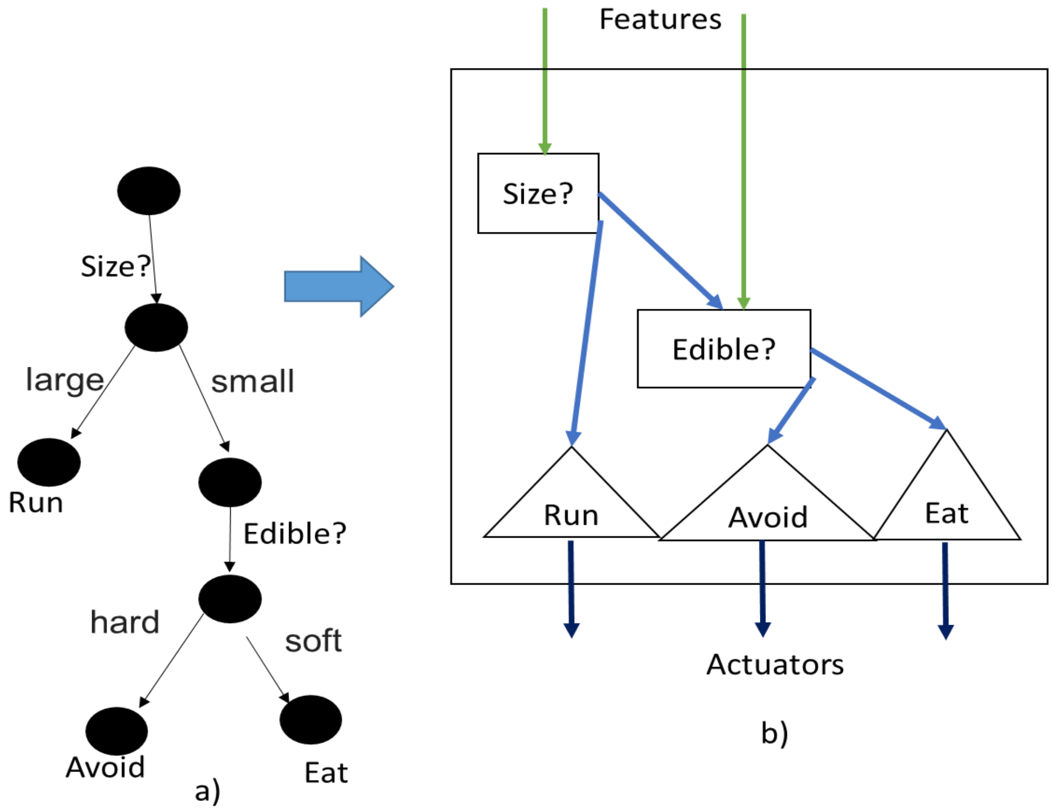

Figure 3: Decision tree iterating the Yes/No decision element and its implementation.

Here, the action of eating is no longer unconditional — it has been restricted to soft objects as determined from a test of edibility with the alternative leading to avoidance of the object. The implementation of such a tree is depicted in Figure 3b with a DEVS network model with components taking the form of magnitude-based binary decision elements and a coupling pattern that is similar to that described for finite state acceptor network realizations Zeigler (2021). We dive deeper into this development in the next section.

3.1 Random Forests

Random Forest classifiers are the products of a popular machine learning approach to training and deploying tools that can predict, for example, whether a smartphone has been infected by malware. The Random Forest technique is an ensemble supervised classifier trained on large feature datasets with the goal of classifying applications as malicious or benign based on performance, memory and other attributes that are easily observed (Alam and Vuong 2013). The detection accuracy depends on free parameters of the Random Forest algorithm such as the number of trees, and depth of each tree and number of random features selected. More detail on Random Forest classifiers can be found in the Wikipedia article https://en.wikipedia.org/wiki/Random_forest. From the DEVS perspective, a forest is a coupled model with decision tree components, such as depicted in Figure 3b, coupled in parallel. The DEVS building blocks and coupling pattern discussed earlier enable DEVS implementation of such coupled models for both simulation and real-time application. Appendix B discusses such an implementation.

3.2 Summary of DEVS Decision Realization Results

The above discussion results in a complexity ranking of decision behavior primitives: 1) First-arrival, 2) Yes/null, and 3) Yes/No. While First-arrival requires only all-or-none event messages, the latter, more complex, elements require event messages with magnitude information. Moreover, the more complex form of such magnitude-based elements requires two distinct output ports for the Yes/No alternatives. Further, in the iterated form of decision tree structure, exemplified by the Random Forest architecture, we need to add activation ports to atomic models so that the output decisions flowing from parent to children are differentially propagated. the Random Forest architecture represents an existence proof for the meaningful application of the DEVS neuromorphic primitives and coupling patterns to implement cognitive behavior.

The use of the hierarchy of system specifications and morphisms in this application is summarized in Table 1.

Table 1: Levels of system specification and morphisms in the DEVS Decision Primitives.

Level	**Specification Name**	**DEVS Decision Primitive**
0	Observation Frame	Input: event message with positive real value Output: event message occurring on one (Yes/null) or two (Yes/No) ports (Figure 2)
1	I/O Behavior	Event occurrence on input may result in output on a port (Figure 2a,b)
2	I/O Function	Event occurrence on input may result in output on a port depending on magnitude relation to threshold (Figure 2a,b)
3	State Transition	Minimal realization of the I/O Function determines the minimum number of states and their transitions/output required to realize the function (Figure 2c,d)
4	Coupled Component	One or more network implementations of the minimal realization with components that are reusable building blocks using architectural coupling patterns (Figure 3)

4 CONTRAST WITH CURRENT THEORIES OF MIND

In the sequel, we attempt to contrast the level of system specification based bridging approach with three approaches to developing a theory of mind representing today's state-of-the-art.

The first approach is that taken by chief founder of Artificial Intelligence, Marvin Minsky's Society of Mind (Minsky 1986). This work is full of insights into how such a theory would work but is not presented in a formal manner. We discuss it here for the complexities that it reveals and the contrast that it suggests with our formal levels of systems specification approach. Mind is conceived as a collection of agencies hierarchically constructed, with agents at the lowest level capable of simple "mindless" functions. Minsky lays out the difficulties encountered in reconstructing cognitive behavior from primitives, emphasizing the combinatorial explosion of interactions: "First we will have to understand how brain cells work. This will be difficult because there are *hundreds of different types of brain cells.* Then we'll have to understand how the cells of each type interact with the other types of cells to which they connect. There could be *thousands of these different kinds of interactions.* Then, finally, comes the hardest part: we'll also have to understand how our *billions* of brain cells are organized in societies." To make progress in such decomposition and reconstruction will require many new theories and organizational concepts that lay out the principles of how societies of parts can accomplish what those parts cannot do separately. He suggests that exploiting knowledge from evolution is essential in that "the more we can find out about how our brains evolved from those of simpler animals, the easier that task will be."

4.1 Contrast with Minsky's Society of Mind

With this as background, Table 2 summarizes Minsky's theory of society of mind along the dimensions of the methodology advocated in Section 1. Some basic contrasts between the two approaches are included in the last column to illustrate aspects of the levels of system specification approach.

Table 2: Contrast of Minsky's approach to society of mind with the proposed methodology.

Aspect of methodology	**Relation to Minsky Society of Mind**	**Contrast with levels of system specification bridging methodology**
Use of system specification hierarchy and associated morphisms	No specific mention is made.	The complexities that Minsky lays out may be better managed with an ordered framework to relate models of agents and agencies at different levels of organization.
Development Method	Attempt to formulate principles of emergence, development, and coordination of agencies – assemblies of parts considered in terms of what they can accomplish as units, without regard to what each of the parts does by itself. Exploit knowledge of evolution to go from simple to complex. One principle of agency interaction is that of limited modularity called "insulation" relative to siblings but not parents.	Focuses on tasks and their accomplishment as opposed to the formulation of I/O behavior and their system realization.
Minimal forms	Attempt to identify agents as opposed to hierarchical collection of agents, called agencies: Any part or process of the mind that by itself is simple enough to understand. Agents occupy one of two states (arousal, quiescent).	Criterion for atomic entities (agents) is subjective (ability to be understood) as opposed to minimality of state realization.
Network construction	As indicated, hierarchical construction involving agents and agencies The architecture of a mind-society must encourage the *formation and maintenance of distinct levels of management by preventing the formation of connections between agencies whose messages have no mutual significance (insulation)*.	Criteria for society of mind architecture as opposed to criteria for well-defined and repeatable simulation compositions.
Model formalism for Simulation and Design	As Minsky indicates, theories need to be specified clearly enough to enable simulation and will require high intensity computation with capacity and speed to simulate enough agents.	Support for model continuity through stages of development including hardware realization viz., neuromorphic architectures.

4.2 Contrast with Approaches of Grossberg and Wang

Grossberg (2021) summarized a long history of research aimed at developing a unified theory that links mind and brain. He outlined the main design principles, mechanisms, circuits, and architectures that show how psychological functions arise as emergent properties of brain mechanisms. As an example, Adaptive Resonance Theory, a core model, shows how advanced brains learn to attend, recognize, and predict objects and events in a changing world that is filled with unexpected events. He also asserts that a theory of this kind may be useful in the design of autonomous adaptive agents in engineering and technology.

A very broad formulation of the problem of bridging cognitive and neuron circuit levels is found under the rubric of cognitive informatics, a discipline concerned with applying information technology and computer science to solve challenges in neuroscience. Wang (2003), a main proponent, presents a set of theories and mathematical models to explore natural and computational intelligence. A wide range of applications of cognitive informatics theory includes cognitive computers, cognitive properties of knowledge, simulations of human cognitive behaviors, cognitive complexity of software, autonomous agent systems, and computational intelligence.

Table 3 compares the approaches of Grossberg and Wang with the levels of system specification methodology along the dimensions identified earlier. In overview, Grossberg and Wang differ fundamentally in their preferred formalisms, with Grossberg committed to systems of differential equations as the only way to construct models while Wang sees mind as an advanced form of software executed in real time. Both construct complex networks using component abstractions that are quite removed from neuronal details. This leaves the gap between cognitive behavior and neuron level circuits open to question. Grossberg's Method of Minimal Anatomies expresses a working approach to model development consistent with accepted philosophical and scientific principles. However, it fails to enjoy the specific structural uniqueness properties that can be inferred from minimal realization theory.

Table 3: Comparison of the approaches of Grossberg and Wang with the proposed methodology.

Aspect of methodology	**Grossberg**	**Wang**
Use of system specification hierarchy and associated morphisms	No specific mention is made.	No specific mention is made.
Development Method	Construct minimal anatomy models that demonstrate evolutionary success; validate via analysis and simulation	Mathematical abstraction, symbolic reasoning, and formalism rigorously model the cognitive process of problem solving in order to enable machine simulation of the human cognitive process.
Minimal forms	Method of Minimal Anatomies - a mathematical model embodies the psychological principles using the *simplest* possible differential equations. The first task of the mathematical model is to explain and predict a lot more behavioral data than were used to derive it.	No specific mention is made.

Network construction	Functional units are not individual cells, but rather whole networks of cells. Equations are used to define a somewhat larger number of modules, or microcircuits that can carry out different functions within each modality (vision, audition, cognition). Modal architectures connect module equations, via informal unspecified coupling of differential equations.	A meta-process serves as a basic building block for modeling software behaviors. Complex processes can be composed from meta- processes using process relational operations. in the context of real-time software behaviors.
Model formalism for Simulation and Design	Asserts that any minimal mathematical model that is capable of realizing proposed design principles in real time requires systems of differential equations	Real-time process algebra (RTPA) provides a coherent notation system and a formal engineering methodology for modeling both software and intelligent systems. RTPA can be used to describe both logical and physical models of systems, where logic views of the architecture of a software system and its operational platform can be described using the same set of notations.

5 DISCUSSION AND CONCLUSION

The hierarchy of system specifications and morphisms affords a framework that supports modeling and simulation in a consistent manner that links knowledge gained at all levels of structure and behavior. This methodology seeks to computationally bridge models at the level of neural circuits and those at the level of cognitive behavior. Such a framework is a family of models and pair-wise mappings that provide a credible path for neural-based mechanistic generation of cognitive behavior. We demonstrated the application of this methodology to extend the set of DEVS building blocks and architectural patterns based on minimal realizations and the levels of system specification developed earlier. We illustrated our approach to establishing minimal realizations of given cognitive behaviors is to DEVS decision behaviors, showing how it helped us to distinguish between Yes/null and Yes/No decision elements and their use within complex decision trees and advanced classifier implementations as DEVS coupled models. Prior work by Mittal and Zeigler (2014) developed an attention-focusing architecture for building intelligent systems including sampling algorithms based on winner-take-all formulated in DEVS. Going beyond the basic building blocks discussed so far, it allows a new sensor activity to break through an existing activity matrix. An interesting open question is how to relate such more complex models to the minimal realizations discussed here. Also, in continued research, we aim to apply this approach to attempts to measure individual neuronal firing patterns and their relation to information coding in network compositions (Pryluk 2019; Ballesta 2022).

Considering the information flows themselves lead us to examine the form of messages sent/received by the components and the corresponding types of the inputs and output ports. As examples of such considerations, neuromorphic hardware implementations of spiking neuron nets have to make such architectural choices in order to tradeoff processing capability with energy consumption and design complexity (Adegbija 2022). The proposed methodology has the potential to support computational tools that enable compositions of well-defined simplification operations from cognitive behavior to neural level realization. Temporal constraints that flow in both directions – in both structure-to-behavior and reverse – are *uniquely* enabled by the DEVS formalism. Current state-of-the-art employs neuromorphic primitives that do not necessarily comply with physiologically based brain network properties/constraints and do not require the simplification stages required when mapping onto cortical/sub cortical networks. As a result they offer limited insight into the way real neural wetware realizes cognitive behaviors. In contrast, we seek to support experimental investigation of cognitive behaviors and modeling them in abstraction stages by biologically plausible networks in brain structures (Feinerman, Rotem and Moses 2008). Moreover, newly available experimentation technologies can be marshalled by transdisciplinary teams of biologists and computer scientists specifically brought together to test simulation models developed through the realization chains derived (Mascart 2021).

A APPENDIX: IMPLEMENTATIONS OF YES/NO WITH HOMOMORPHISM PROOF

Figure 4 shows a DEVS coupled model in MS4 Me (Seo et al. 2013) with two components to realize the binary choice Yes/No. One component realizes the Yes/null behavior associated with a single neuron, the other adds the needed No output functionality. The input X port is coupled to both the input X port of the Yes/No atomic model as well as the Activate port of the No atomic model. The latter sends the model to the Active state where it is scheduled to stay for a duration by its time advance (here 10). At the expiration of this time, a No output would be generated unless a DeActivate input sends the model back to its initial state. This input is generated only if the input crosses the threshold and outputs a DeActivate message simultaneously with the Yes output.

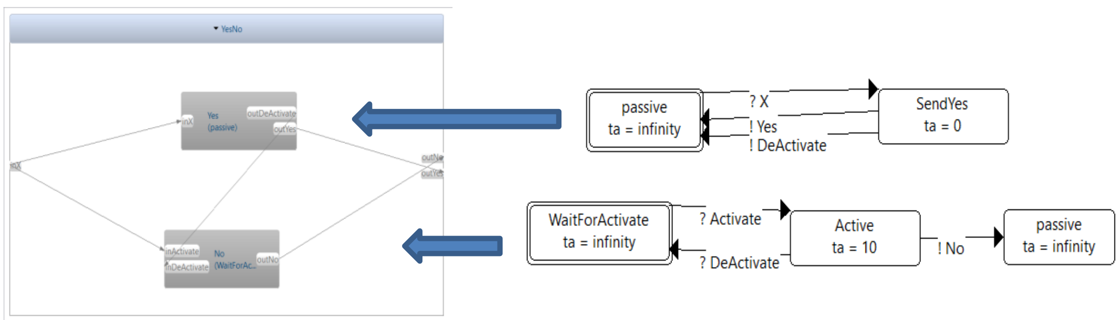

Figure 4: Realization of the Yes/No behavior using a DEVS coupled model with two components.

To verify that the coupled model simulates the atomic model and therefore has the same behavior, we construct a mapping as defined in Table 4.

Table 4: Mapping from Coupled to Atomic Realization of Yes/No Behavior.

Model/State	Coupled Yes/No	Atomic Yes/No	Output
1	(passive,WaitForActivate)	pass	null
2	(sendYes,Active)	sendYes	Yes
3	(passive,Active)	sendNo	No

To establish the mentioned simulation relation requires showing that the mapping is preserved under state transitions caused by all events both external and internal. For example, the state transition from state 1 to state 2 in the table is caused by the external input of an above threshold input in both models. We omit the technical details and refer the reader to (Zeigler, Muzy and Kofman, 2018).

B APPENDIX: IMPLEMENTATIONS OF RANDOM FOREST

Figure 5a sketches a decision tree that constitutes a component in a Random Forest (Alam and Vuong 2013).

In a DEVS coupled model implementation, Yes/No atomic models populate the interior nodes, while simple atomic models generate the outputs at the leaves. Each input feature connects to one or more interior nodes which tests its value against a threshold as implemented by the node's Yes/No model.

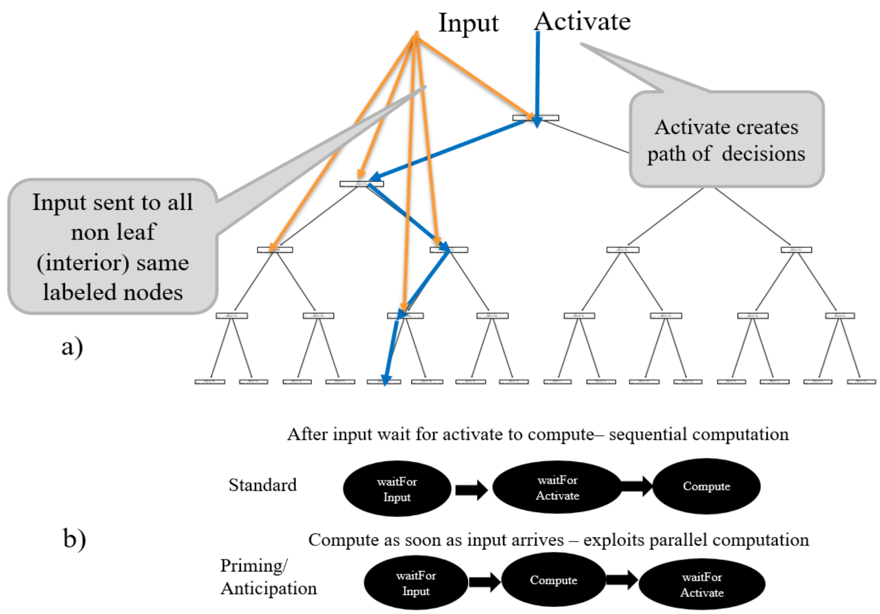

Figure 5: DEVS implementations of a decision tree that constitutes a component in a Random Forest.

The assignment of input features and thresholds to nodes constitute the parameters of the tree and are determined in training against I/O data. As illustrated in Figures 5a,b, an Activate message is transmitted from parent to the child determined by threshold-based decision. The path of activation from root to leaf constitutes the decision computation of the tree. Two implementations are possible. In the standard form. after receiving the external input, nodes wait to be activated by their parent and proceed to compute the threshold comparison. In the second form, after receiving external input, all nodes compute the threshold comparison concurrently. This is possible because the threshold only depends on the input and the preassigned computations that are not on the activation path are "wasted". Nodes then wait for parental activation before propagating their own activation. In this form, they take the maximum time of any one. In the standard form, computations are only made on the activation path but the sequential process takes the sum of the individual times on this path. In brief, let the depth of a tree = *d*, the node time of computation $= t_c$, and the node time of activation $= t_a$. Then the computation time for the standard (sequential) implementation is $T_S = d(t_c + t_a)$, and the computation time for the Parallel implementation is $T_P = t_c + dt_a$, because all computations are performed concurrently. The speedup is then $T_S/T_P = \sim d$ where the node activation time is considered negligible compared to the computation time. The speedup of a forest is governed by the deepest tree if all of the trees report their decisions to generate the final output. This is a special case of sequential versus parallel simulation discussed by Zeigler (2017). A homomorphic proof of simulation similar to that of Appendix A is used to verify correct implementation of each processing method.

REFERENCES

Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2000. *Discrete-Event System Simulation*. 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Adegbija, T. 2022. "DINGO: A Modeling Approach for Translating Spiking Neural Networks to Hardware Accelerators". *Submitted to IEEE Transactions on Systems, Man, and Cybernetics*.

Alam, M.S., and S.T. Vuong. 2013. "Random Forest Classification for Detecting Android Malware". In *Proceedings of the IEEE International Conference on Green Computing and Communications*, August 20th-23rd, 663-669.

Ballesta, S. Weikang, K.E. Conen, and C. Padoa-Schioppa. 2022. "Values Encoded in Orbitofrontal Cortex Are Causally Related to Economic Choices". *bioRxiv preprint*.

Carandini, M. 2012. "From circuits to behavior: a bridge too far?". *Nature Neuroscience* 15(4):507-509.

Castro, S., P.J. Mosterman, A.H. Rajhans, and R.G. Valenti. 2019. *Challenges in the Operation and Design of Intelligent Cyber-Physical Systems*. In Complexity Challenges in Cyber Physical Systems: Using Modeling and Simulation (M&S) to Support Intelligence, Adaptation and Autonomy. Wiley: Hoboken, NJ, USA.

Douglass S., and S. Mittal. 2013. "A Framework for Modeling and Simulation of the Artificial. In *Ontology, epistemology, and teleology for modeling and simulation* 44:271-317.

Feinerman, O., A. Rotem, E. Moses. 2008. "Reliable neuronal logic devices from patterned hippocampal cultures". *Natural Physics* 4:967–973.

Gautrais, J., S. Thorpe. 1998. "Rate Coding Versus Temporal Coding: A Theoretical Approach". *BioSystems* 48:57–65.

Gigerenzer, G., D.G., Goldstein. 2000. "Reasoning the Fast and Frugal Way: Models of Bounded Rationality". In *Judgment and Decision Making*, edited by T. Connolly, 621–650. Cambridge, UK: University Press.

Grossberg, S. 2021. *Conscious Mind/Resonant Brain: How Each Brain Makes a Mind*. New York, NY: Oxford Univ. Press,

Krohn, K., J. Rhodes. 1965. "Algebraic theory of machines. I. Prime decomposition theorem for finite semigroups and machines". *Transactions of the American Mathematical Society* 116:450-464.

Mascart, C., G. Scarella, P. Reynaud-Bouretb, A. Muzy. 2021. "Simulation scalability of large brain neuronal networks thanks to time asynchrony". *Biorxiv* .

Minsky, M. 1986. *The Society of Mind*. Simon and Schuster. NY.

Mittal, S., B.P. Zeigler. 2014. "Modeling Attention Switching in Resource Constrained Complex Intelligent Dynamic System (RCIDS)". In *Proceedings of the Symposium on Theory of Modeling and Simulation-DEVS*, April 13th-16th, Tampa, Florida, 1-10.

O'Connell, R.G., M.N. Shadlen, K. Wong-Lin, S.P. Kelly. 2018. "Bridging Neural and Computational Viewpoints on Perceptual Decision-Making". *Trends Neuroscience* 41(11):838-852.

Petersen, S. E., O. Sporns. 2015. "Brain Networks and Cognitive Architectures". *Neuron* 88(1):207-219.

Pryluk R., Y. Kfir, H. Gelbard-Sagiv, I. Fried, R. Paz. 2019. "A Tradeoff in the Neural Code across Regions and Species". *Cell* 176(3):597-609.

Seo, C., B.P. Zeigler, R. Coop, D. Kim. 2013. "DEVS Modeling and Simulation Methodology with MS4Me Software TMS". In *Proceedings of the 2013 Spring Simulation Multiconference*, April 7th-10th, San Diego, CA, USA, 7–10.

Stearns, R. E., J. Hartmanis. 1966. *Algebraic Structure Theory of Sequential Machines*. United States: Prentice-Hall.

Vahie, S. 2000. "Dynamic Neuronal Ensembles: Neurobiologically Inspired Discrete Event Neural Networks". In *Discrete Event Modeling and Simulation Technologies*. Sarjoughian, H., Ed. Springer: Berlin/Heidelberg, Germany.

Wang, Y. 2003. "On Cognitive Informatics". *Brain and Mind* 4:151–167.

Wong K.F., X.J. Wang. 2006. "A recurrent network mechanism of time integration in perceptual decisions". *Journal of Neuroscience* 26(4):1314-1328.

Zeigler, B.P. 2017. "Using the Parallel DEVS Protocol for General Robust Simulation with Near Optimal Performance". *Computing in Science & Engineering* 19(3):68-77.

Zeigler, B.P. 2020. "Hybrid Iterative System Specification of Cyberphysical Systems: Neurocognitive Behavior Application". In *Proceedings of the 2020 Spring Simulation Conference (SpringSim)*, May 18th–21st, San Diego, CA, USA, 1-12.

Zeigler, B.P. 2021. "DEVS-Based Building Blocks and Architectural Patterns for Intelligent Hybrid Cyberphysical System Design". *Information* 12(12):531.

Zeigler, B.P., A. Muzy, E. Kofman. 2018. *Theory of Modeling and Simulation: Discrete Event Iterative System Computational Foundations*. Academic Press: New York, NY, USA.

ACKNOWLEDGMENTS

I would like to than Dr. Sang Won Yoon, Professor, and Christian Koertje, Graduate Research Associate at the Watson Institute for Systems Excellence at The State University of New York at Binghamton, NY, for their help in working on the Random Forest example.

AUTHOR BIOGRAPHY

BERNARD P. ZEIGLER is Professor Emeritus of Electrical and Computer Engineering at the University of Arizona, Tucson. He is internationally known for his contributions in modeling and simulation theory. He develop the Discrete Event System Specification (DEVS) formalism in 1976. His email is: berniezeigler@gmail.com.