

CLOSURE UNDER COUPLING: CONCEPT, PROOFS, DEVS RECENT EXAMPLES (WIP)

Bernard P. Zeigler
University of Arizona and RTSync Corp.
Phoenix AZ, USA
zeigler@rtsync.com

Key words: Closure under coupling, DEVS, routed DEVS, Min-Max DEVS, DEVS ML

ABSTRACT

With the growth in new variants of DEVS, the concept of closure under coupling has reached a level of importance where it stands discussion in its own right. As emphasized in (Zeigler et al 2000), closure under coupling justifies hierarchical construction. Here we show that it also provides assurance that the class under consideration is well-defined and enables checking for the correct functioning of feedback coupled models. Absence of closure is also informative as it begs for characterizing the smallest closed class that includes the given class. This illustrated here as we discuss closure under coupling for several recently introduced subclasses of the DEVS formalism.

Keywords: Closure under coupling, hierarchical construction, DEVS, RoutedDEVS, Min-Max-DEVS, ML-DEVS, DEVS Markov models

1 INTRODUCTION

The concept of closure under coupling is framed within general dynamic systems as involving a basic subclass of systems such as DEVS, DTSS, and DESS. The case of DEV&DESS (Zeigler et al 2000) brought up interesting complexities. In showing that DEV&DESS it is closed under coupling, Praehofer (1991) considered a) the pairs of input-output interfaces between the different types of included systems, b) the means to specify types of components with intermingled discrete and continuous expressions, and c) an abstract simulator to establish that the new formalism could be implemented in computational form.

Referring to Figure 1, as with the just mentioned system specifications, a basic formalism (1) is assumed that specifies (2) the subclass of basic systems. Also a coupled subclass (3) is assumed that has a specification (4) that is based on coupling (5) of basic specifications. The coupled specification is assumed to produce a resultant (6) which is a coupled system of basic component systems. Closure under coupling for the subclass of interest requires that all such resultants are behaviorally equivalent to basic systems – shown as that the coupled subclass is included within the basic systems subclass. A proof of such inclusion can be had by constructing, for any such resultant, a basic specification (7) of a system equivalent to the resultant. Such a specification can then participate as a component in a coupled system of basic components leading to hierarchical construction that remains within the formalism.

For DEVS, we have a structure M (1) that specifies (2) a System, S_M and a Network structure, N (3) that, based on a coupling recipe (5), specifies (4) a resultant System S_N (6). A DEVS M_N is constructed from the structure of N which specifies a System that is behaviorally equivalent to S_N . Thus any valid coupling of a basic DEVS components results in a system equivalent to a basic DEVS – showing closure under coupling.

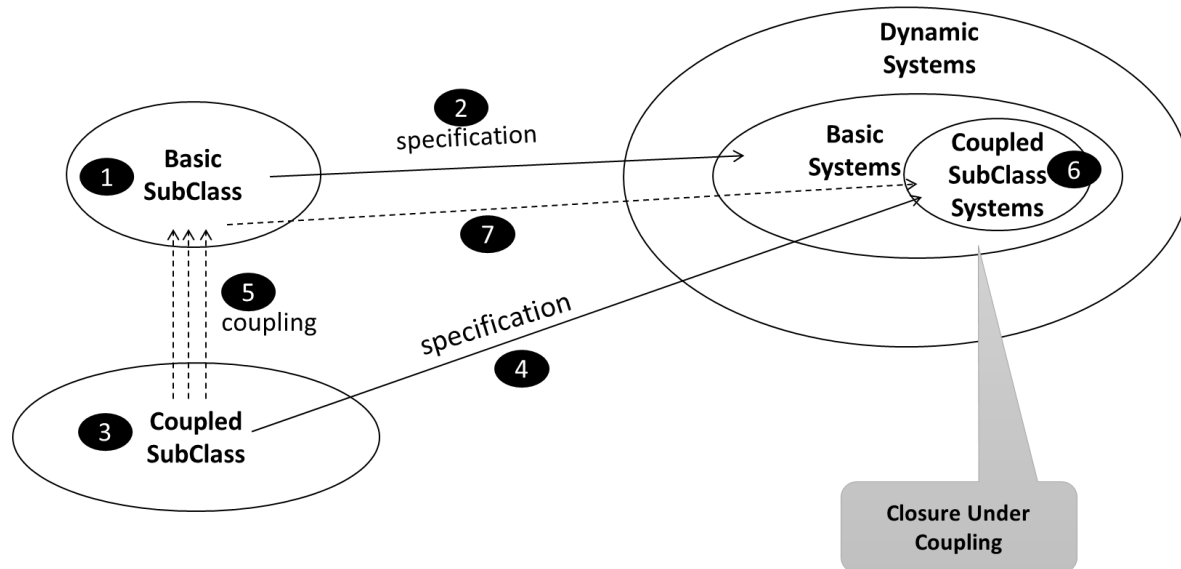


Figure 1. Closure under coupling

Closure under coupling justifies hierarchical construction and flattening from coupled to atomic models (Muzy and Nutaro 2005) as well as assuring that the class under consideration is well-defined, enabling checking for the correct functioning of coupled models (Baohong 2007). Besides closure under coupling, two types of questions arise for such formalisms: 1) are they subsets of DEVS, behaviorally equivalent to DEVS but more expressive or convenient, or bring new functionality to DEVS, and 2) have simulators been provided for them to enable verification and implementation? Here we discuss three examples of recent DEVS-based formalisms where these questions arise. Overall, we will see that such proofs can be exercises in reducing the additional functionality to that available in DEVS itself by expanding the state set sufficiently to support their explicit operation in DEVS. Besides supporting hierarchical construction such exercises can push the designer toward well-definition of the formalism itself in an iterative development process. In the rest of the paper, we discuss a variety of examples from recent DEVS literature that either consider closure under coupling explicitly (RoutedDEVS and Multi-Level DEVS) or do not do so (Min-Max-DEVS). The former offer instances to examine the significance of the concept and its proof in context as well as to generalize on these issues. The latter offers an opportunity to address what might be missing in the formalism as presented and what benefits might be derived from considering the closure under coupling property for the introduced formalism. Absence of closure is also informative as it begs for characterizing the smallest closed class that includes the given class. This illustrated here as we discuss closure under coupling of a fourth example, DEVS Markov classes.

The first example, Multi-Level DEVS references the original paper by Steniger and Uhrmacher (2016) to point to a complete proof of closure under coupling. The second example points out that although a network of Routed DEVS [Blas et al 2017] components is a well-defined DEVS, closure under coupling requires proof that it is equivalent to a basic Routed DEVS. The third example Min-Max DEVS (Hamri et al 2006) illustrates how consideration of closure under coupling can bring up new questions concerning the definition of the formalism.

2 EXAMPLE: MULTI-LEVEL DEVS

The Multi-Level DEVS (ML-DEVS), based on Parallel DEVS, follows the reactive systems metaphor with particular application to computational biology [Uhrmacher et al. 2007]. It supports modeling combinations of different levels of organization and the explicit description of interdependencies between those levels. ML-DEVS supports MICRO-DEVS (atomic) and MACRO-DEVS (coupled) models, respectively. Different from DEVS, MACRO-DEVS coupled models have a state and behavior of their own thereby obviating an additional atomic model that otherwise is needed to represent the macro-level behavior. A recent extension (Steniger and Uhrmacher, 2016) adds an expressive intensional (i.e., implicit) coupling mechanism which employs dynamic structure to selectively enforce explicit couplings depending on the state of the model (see also Park and Hunt, 2006). A formal proof that ML-DEVS is closed under such coupling shows that a MACRO-DEVS model can be specified as a MICRO-DEVS model (i.e., a basic model in the formalism as discussed above.) So to construct a MICRO-DEVS model that is behaviorally equivalent to any given MACRO-DEVS, Steniger and Uhrmacher, (2016) must take account of the additional structure at the Macro-DEVS level. In turn, this requires considering “vertical” couplings expressing upward and downward causation between micro and macro levels as well as the usual “horizontal” couplings in DEVS coupled models. It also requires accounting for the intensional coupling mechanism, as well as other features such as interfaces that can belong to components, active and inactive ports, and accessible and private states. The reader is referred to the Appendix of (Steniger and Uhrmacher, 2016) for details of this proof which shows explicitly how the resultant’s (qua basic model) state and characteristic functions are defined to account for all the behavior introduced in its coupled model incarnation.

3 EXAMPLE: ROUTED DEVS

Routed DEVS (RDEVS) is a subclass of DEVS intended to hide the handling of routing information among DEVS component models in a DEVS network [Blas et al 2017]. The RDEVS formalism is based on DEVS and defines a set of DEVS atomic models that use routing information to authenticate senders, receivers, and transactions. Interpreting this incoming information, routing models only accept specified input messages before passing on the message payload to an associated model for processing. Also, routing models are capable of directing processing results to a specific set of receivers. Blas et al [2017] define the associated model (called essential model) as embedded within the routing model. A RDEVS Network is defined with all-to-all coupling of routing models. Note that although the network is a well-defined DEVS coupled model based on DEVS closure under coupling, it is not necessarily equivalent to a routing model. Hence, closure under coupling of RDEVS shows how to define an equivalent routing model for any such network. This requires that the structure of a routing model is extracted from the network structure and shown to be behaviorally equivalent to it.

To illustrate the closure under coupling proof we adopt the representation of a routing model in Figure 2 a).

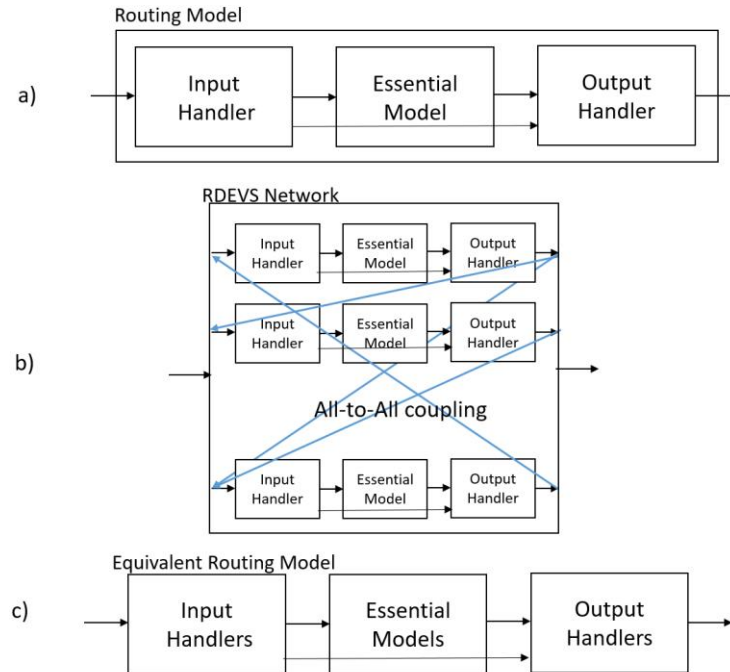


Figure 17. RDEVS formulation to illustrate closure under coupling

Here the Input and Output Handlers take care of the routing information in the incoming message and pass on the operative content to the Essential Model for processing. The coupling from the Input to Output Handler is for passing on the authentication and routing information which depends on the processing state. The Routing Network is then a coupled model of Routing Models with all-to-all output to input coupling in 2b). The equivalent Routing Model of the Routing Network is then constructed first by defining equivalent models for the groups of Input, Output Handlers and Essential Models. Then these are coupled as in 2c) to define the equivalent Routing Model of the Routing Network. One advantage of this approach is that all models are explicit and not embedded in others. Of course the original may be faster in processing. The construction of the equivalent routing model for any routing network demonstrates closure under coupling of the RDEVS formalism and therefore hierarchical construction that stays within the formalism. This allows the use of the RDEVS formalism as a “layer” above the DEVS formalism that provides routing functionality without requiring the user to “dip down” to DEVS itself for any functions.

4 EXAMPLE: MIN-MAX DEVS

Min-Max DEVS introduces the concept of ambiguous delay into the DEVS formalism by extending the time advance function to yield an interval (t_{min} , t_{max}) that represents all that is knowable about the timing of the associated state transition (Hamri et al 2006). Other approaches used in digital circuits to represent the imprecise knowledge of real gate delays are based on probabilistic models and fuzzy-logic models and involve complex simulations. However, they do not necessarily give more information than in the min-max approach because the probability or possibility distributions are generally unknown. The extended DEVS atomic state keeps duplicates of the state of the modeled system so that it can track both slow and fast evolution of the system simultaneously. In addition it has a state variable that takes on values: fast, slow, autonomous, and passive that helps make transition decisions. The basic defining rule is: fast events are applied to the faster model (minimum delay) and slow events are applied to the slower model (maximum delay). In order to avoid a combinatorial explosion, the totally unknown state is introduced, in which the model remains forever once entered. The Min-Max-DEVS simulator carries out a single simulation which is equivalent to multiple simulation passes in which the lifetime of transitory states corresponds to all the possible values of the real system. Min-Max-DEVS is still in a state of development so that coupled models

and consequently, closure of the formalism under coupling have not been addressed. However, the basic model is a full-fledged DEVS (with input and output sets and functions) and therefore can support coupling into network models. As discussed above the proof of closure under coupling would present interesting challenges: Does the state of a coupling of Min-Max-DEVS components define a well-defined time advance of the form (t_{min}, t_{max}) ? Can such a global state, whose components each contain the transition decision support variable, be mapped to a single variable of the same kind?

5 EXAMPLE: DEVS MARKOV MODELS

Markov Modeling is among the most commonly used forms of model expression and many classes have been studied, including Markov Chains (Feller 1966), Continuous Time Markov (CTM) Processes (Kemeny and Snell 1960), Discrete Time Markov (DTM), Semi-Markov Processes (Barbu and Limnios 2008), Generalized Semi-Markov Processes (Glynn 1989; Younes and Simmons 2004), Markov Decision Processes (Rachelson, E. et al. 2008; Puterman 1994), and Hidden Markov Models (Barbu and Limnios 2008). The Stochastic DEVS formalism was defined by Castro et al (2010) to provide a stochastic extension of the DEVS formalism. Coupled models with Stochastic DEVS components are taken to be defined in the usual DEVS form. The proof given by Castro et al. (CSW 2010) that Stochastic DEVS is closed under coupling showed that they defined Stochastic DEVS in a way that is consistent with underlying fundamental probability theory. This follows this from the fact that the resultant of coupling must have same probability structure as that underlying the basic model.

DEVS Markov Models (Seo et al. 2018, Zeigler et al 2018) form a subclass of Stochastic DEVS and can represent complex systems at the level of individual subsystems and actors. In this guise, each system or actor can be represented as a component with states and transitions as well as inputs and outputs that enable it to interact as atomic models within coupled models using coupling in the usual way. The DEVS Markov class, which assigns both probabilities and transition time distributions to state transitions, was shown to be closed under coupling. However, the smaller class DEVS Continuous Time Markov (CTM), which only assigns probabilities to state transitions, is not closed under coupling. Further, closure of the DEVS CTM class can be proved when dependence on elapsed time is not allowed. In this instance, the global states of couplings of CTM components depend only on the current sequential states of their components, not their total states (which include elapsed times). This allows the sequential states to serve as elements of the global Markov state required for closure under coupling (Zeigler, Nutaro, and Seo, 2017). Closure under coupling provides assurance that the class under consideration is well-defined and enables checking for the correct functioning of coupled models. Absence of closure is also informative as it begs for characterizing the smallest closed class that includes the class. For the DEVS CTM class, it can be shown that the smallest such class is the DEVS Markov class itself (Seo et al. 2018).

6 CONCLUSION

With the growth in new variants of DEVS, the concept of closure under coupling has reached a level of importance where it stands discussion in its own right. Besides justifying hierarchical construction and flattening for efficient simulation, closure under coupling also provides assurance that the class under consideration is well-defined and enables checking for the correct functioning of feedback coupled models. Absence of closure is also informative as it begs for characterizing the smallest closed class that includes the given class. Here we discussed a variety of examples from recent DEVS literature that offer vehicles to examine the significance of the concept and its proof in context as well as to generalize on these issues. Lack of closure offers an opportunity to address what might be missing in the formalism as presented in the literature and what benefits might be derived from considering the closure under coupling property for the introduced formalism.

REFERENCES

- Baohong, L. 2007, A Formal Description Specification For Multi-Resolution Modeling Based On DEVS Formalism And Its Applications, *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology* July 2007 vol. 4 no. 3 229-251
- Blas, MJ, S. Gonnet, and H. Leone, 2017 Routing Structure over Discrete Event System Specification: A DEVS Adaptation To Develop Smart Routing In Simulation Models, *Proceedings of the 2017 Winter Simulation Conference* W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds, pp. 774-785
- Park, S., Hunt, C. A., 2006, Coupling permutation and model migration based on dynamic and adaptive coupling mechanisms. In: *The Proceedings of the 2006 DEVS Symposium*
- Hamri, E.A., N. Giambiasi, C. Frydman, Min–Max-DEVS modeling and simulation, *Simulation Modelling Practice and Theory* 14 (2006) 909–929
- Muzy, A. and JJ Nutaro. Algorithms for efficient implementations of the devs & dsdevs abstract simulators. In *1st Open International Conference on Modeling & Simulation (OICMS)*, pages 273–279, 2005
- Park, S., Hunt, C. A., 2006, Coupling permutation and model migration based on dynamic and adaptive coupling mechanisms. In: *The Proceedings of the 2006 DEVS Symposium*.
- Praehofer, H. 1991. *System Theoretic Foundations for Combined DiscreteContinuous System Simulation*, Doctoral dissertation, Johannes Kepler University of Linz, Linz, Austria.
- Seo, C., Zeigler B.P., and Kim D.H, 2018 DEVS Markov Modeling and Simulation: Formal Definition and Implementation, *SpringSim 2018*, April 15-18, Baltimore, MD, USA; DEVS markov models
- Steniger A. and A. Uhrmacher, 2016 Intensional Coupling in Variable Structure Models: An Exploration Based on Multi-level DEVS, *TOMACS*, Volume 26 Issue 2.
- Uhrmacher, A.M. et al. 2007. Combining micro and macro-modeling in DEVS for computational biology. In *Proceedings of the 2007 Winter Simulation Conference*, S. G. Henderson, B. Biller, M.-H. Hsieh, J.
- Zeigler, B.P., Praehofer, H., and Kim, T.G.. 2000. *Theory of Modeling and Simulation.*, Academic Press Publisher, NY.
- Zeigler B.P., Nutaro, J., and Seo, C., 2017, “Combining DEVS and Model-Checking: Concepts and Tools for Integrating Simulation and Analysis,” *International Journal of Simulation and Process Modelling (IJSPM)*, Vol. 12, No. 1, doi.org/10.1504/IJSPM.2017.082781
- Zeigler, B.P., Muzy, A., and Kofman, E. 2018. *Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations*, 3rd Ed., Elsevier Publisher, NY.

AUTHOR BIOGRAPHIES

Bernard P. Zeigler is Chief Scientist at RTSync Corp., Professor Emeritus of Electrical and Computer Engineering at the University of Arizona, Tucson and co-Director of the Arizona Center for Integrative Modeling and Simulation. His email address is zeigler@rtsync.com.