# Bridging Modeling Methods: A Formalism to Specify Hybrid Simulation Models on the Conceptual Level

**Joachim Block**
archibald@ieee.org

In the last years, parallel to the ever-growing body of knowledge in SD, we have recognized an increasing interest in combining SD with other simulation paradigms, including, discrete event simulation (DES) and agent-based modeling and simulation (ABMS). The appearance of multi-method tools such as AnyLogic or Nova has reinforced this common trend toward hybrid modeling and simulation.

Although pure SD is mostly preferable, a hybrid approach may have advantages in certain situations. For instance, hybrid models can address problems from different dimensions and, hence, can offer a more comprehensive and holistic view. Real-world problems are complex, and mixing paradigms is one way to capture this complexity in a single model. Due to conceptual limitations, a class of systems exists that is difficult or, in some cases, even impossible to model using one paradigm alone.

SD and ABMS can complement each other purposefully. By overcoming their inherent limitations simultaneously, each paradigm's strengths can be exploited. In particular, both the macro and the micro levels of a real-world problem can be addressed. For instance, hybrid workforce planning and healthcare models show the benefits of a hybrid approach.

However, a commonly accepted formalism for specifying hybrid models on the conceptual level - such as SD stock-and-flow diagrams - is still missing. The conceptual model is the most crucial step in the modeling and simulation cycle. It bridges two worlds: the real-world problem and the implementation of a simulation model.

Our hybrid agent-based modeling formalism - the Hybrid Agent-Based Modeling (HABM) formalism - contributes to filling the existing gap. The HABM formalism enables us to specify agent-based models enriched with elements from DES and SD. The formalism enhances communication between different model stakeholders, such as problem owners and implementation team members, and it also provides a platform for reusing pre-specified model components. In addition to integrating ABMS, DES, and SD, dynamic model structures are also considered.

The HABM formalism is, in general, founded on hybrid automaton theory. This stream of research concerns systems exhibiting continuous behavior disturbed by discrete events. We orient, to a great extent, on the discrete event and differential equation system specification (DEV&DESS) formalism. DEV&DESS has found a well-established place within the modeling community. It supports, among others, the specification of both static and dynamic networks of hybrid automata.

The core definition of the HABM formalism is about entities representing both agents and objects with the latter forming the environment of the agent population. Every entity can have a rich internal structure with two sub-systems defined by SD and DES. Figure 1 illustrates the example of an HABM entity.

A labeled and directed state transition graph describes the discrete state sub-system (DES).. A set of discrete states, forming the nodes of the graph, are connected with directed edges $E$. An edge $(q,q') \in E$, with $q,q' \in Q$, means that a transition from $q$ to

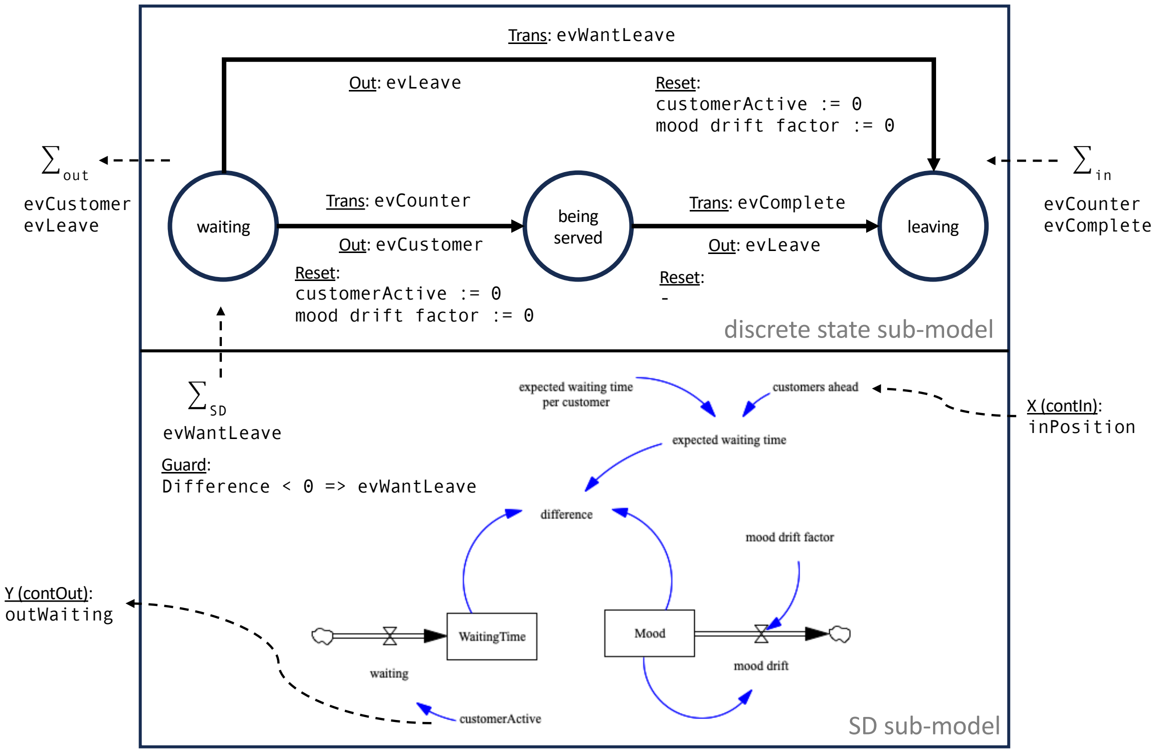


Figure 1: A customer in a waiting line model specified as an HABM entity

$q'$ is possible. Every edge is labeled with a set of events (*Trans*), a set of output events (*Out*), and a *Reset* funtion. A state transition is performed if one of these events occurs. Events arrive from the external environment or are created by the internal dynamics.

The continuous state sub-system is defined by a SD model. It receives continuous input from and sends continuous output to the environment. This sub-system also sends discrete events to the discrete state transition sub-system. A *Guard* function evaluates all continuous variables and triggers an internal event when certain thresholds are exceeded. This function not only considers the continuous values but also takes the current state of the discrete state sub-system into account.

The dynamics of the entity are as follows. An event forces the discrete state sub-system to perform a state transition (*Trans*). Before the transition is performed, an output event is created and sent to the external environment (*Out*). Furthermore, the continuous state sub-system is reset accordingly (*Reset*) forcing a disruption of the SD mdoel.

As agents and objects do not stand in isolation, HABM offers the definition of a (static structure) network. Conform to DEV&DESS, interface maps are introduced to couple HABM entities. However, in most hybrid models with elements of ABMS, more than a static structure network is required. Links between agents often change as time elapses.

The specification of dynamic structure HABM networks is realized by introducing a specific model element: the network executive. This dedicated network element has encoded the entire network as a state variable. Functions in the network executive control and adapt the network during runtime.

The HABM formalism supports the specification of hybrid models with elements from SD, DES, and ABMS. One of the main limitations is the specification of the network executive's internals. These are specified to some extent by non-standardized macro code.