

Research on Model Composition for Underwater Warfare Simulation in Engagement-Level

Jianchun Zhang^{1, 2, a*}, Fengju Kang^{1, 2, b}

¹Marine College, Northwestern Polytechnical University, Xi'an, 710072, China

²National Key Laboratory of Underwater Information Process and Control, Xi'an, 710072, China

^a511604709@qq.com, ^bkangfengju@nwpu.edu.cn

Keywords: Model Composition, SES, HDEVS, Underwater Warfare, Ontology.

Abstract. This article introduces a structural modeling approach and a rule-driven model composition method for underwater warfare simulation to facilitate the reusability and composability. First, a SES-based model structure is built and the Hybrid Discrete Event System specification (HDEVS) formalism is proposed to model the combat entity. Utilizing the ontology, we describe the designed models in the Underwater Warfare Simulation Ontology (dubbed UWSO) to enhance the readability and disambiguation. Moreover, rules of model composition are performed in syntactic, semantic and pragmatic layers during the integration phase according to certain scenario. Finally, by preliminarily applying in underwater warfare system, the result shows efficiency of the research.

Introduction

As modeling and simulation (M&S) has been widely utilized in the military communities, the various developed combat models have contributed to assessing performances, analyzing tactics, and planning acquisitions of underwater warfare [1]. In spite of their past contributions, the existing models suffer mainly from low model reusability and extendibility. When applying these models to new weapon design or tactical analysis, we are hard to modify, integrate, and reuse the existing models, which results in large costs for the adaptation to new application scenarios. Creating simulation models by composition of predefined and reusable components is a way to reduce the costs and time. The focus of this article, therefore, is to introduce composable methods for a more reusable and extendable underwater warfare modeling and simulation.

We propose the overall model structure to understand the model construction and implied relationships between upper models and their components using system entity structure (SES). This model structure enables one to express the real system more effectively and to reflect extendable underwater warfare scenarios. Then using the DEVS formalism, it enables a more straight-forward model comparison and the bigger model development with less cost. Aiming at overcoming disadvantages and benefiting from the merits of hybrid simulation, we adopt a representative approach called HDEVS [2], which consists of hybrid coupled models, DEVS coupled models, DEVS atomic models, continuous coupled models, and continuous atomic models. This formalism emphasis, however, means users should develop separate models, not an integrated model from a hybrid system and these separate models cannot reflect the original form of a target system. Hence, a modified HDEVS formalism to model the combat entity is proposed and the key improvement is the unification of definition, known as hybrid coupled models (HCMs) and hybrid atomic models (HAMs). On the basis, UWSO is proposed to describe concepts and relationships, which formally represents and shares domain knowledge as a set of concepts to improve reusability and interoperability [3].

Composability has been defined as “the capability to select and assemble reusable simulation components in various combinations into simulation systems to meet user requirements [4,5]”. Syntactic composability is concerned with the compatibility of implementation details [6,7]. Semantic composability, on the other hand, is concerned with whether the models that make up the composed simulation system can be composed in a meaningful way. To enrich the elements of composition and promote simulation system development, it needs to build more complete rules to guide the composition process.

This article contains five sections. Section 2 introduces the modeling approach using SES and HDEVS and ontological description method, and Section 3 explain the proposed a model composition process with detailed rules. Section 4 illustrates the experiment to prove the efficiency of this work. Finally, Section 5 concludes this study and proposes future extensions for a more complete solution.

Model Design

Overall Model Structure. To systematically organize the family of models into a repository, we use the system entity structure (SES) to describe the overall model structure for underwater warfare, which offers a structural means of knowledge representation. The model repository characterizes taxonomic and decomposition relationships among component models to help compose the right combinations of alternatives to satisfy the particular objectives, as illustrated in Fig. 1.

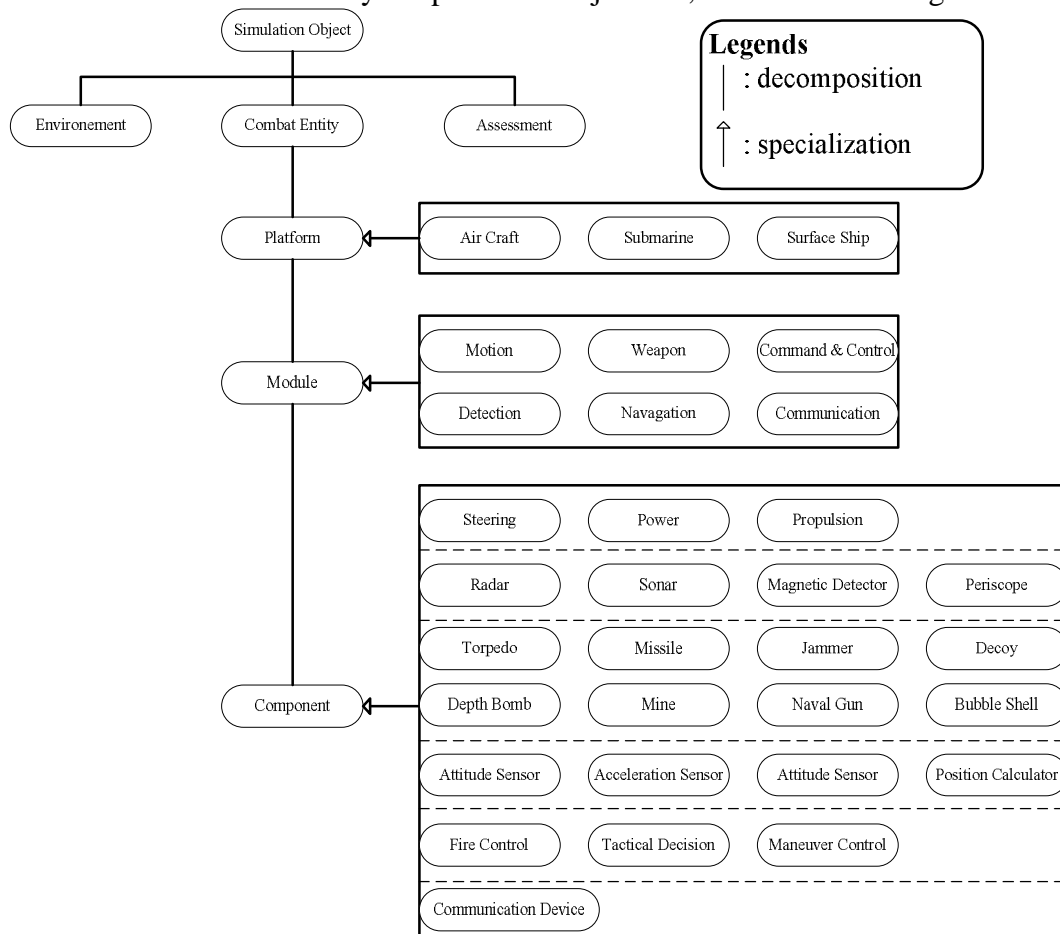


Fig. 1. Overall model structure of underwater warfare using SES

Concretely, the underwater warfare simulation model, an uppermost model, is divided basically into environment, combat entity and assessment models that we are interested in modeling. The combat entity models are described in three levels that include platform, module and component; the environment model reflects environmental effects such as weather patterns, temperature, salinity, artificial features and more; the damage assessment model evaluates the engagement situation (e.g., the offensive combat entity model attacks the target or the defensive model defends against the attack);. In spite of the brief model construction using SES, the proposed model structure assures flexibility and extendibility to various underwater warfare scenarios by hierarchical decomposition of focused objects and reduces corresponding system complexity. Especially, some equipment such as torpedo, missile can also be decomposed into several components and viewed as a special platform in simulation domain.

HDEVS Formalism for a Combat Entity. HDEVS is based on DEVS formalism [8]. Each model has I/O ports and is interconnected through coupling relations. Models have ports for two types of data:

digital signals (representing continuous signals) and discrete events. When signals or events are need to be converted, the port mapping functions convert the data. Digital-to-Event (D/E) mapping functions are used for detecting state events triggered when a continuous signal reaches a threshold level. Event-to-Digital (E/D) mapping functions generate mapped signals when a related event occurs.

A focus of the proposed formalism is the extension of atomic and coupled models, as shown in Fig. 2. The modeling is difficult because of the requirement about complex knowledge background: that is, the domain knowledge as well as the M&S knowledge at the same time. To be specific, each model is modeled into two levels in terms of a layered structure through model abstraction: (1) a logical model (LM) layer for the M&S knowledge; (2) a physical model (PM) layer for the domain knowledge. The LM layer represents the abstract behavior of a PM, and it is suitably employed to describe models macroscopically. For microscopic modeling, we adopt the PM layer to reflect detailed behavior of the same model by actions represented as various functions or algorithms.

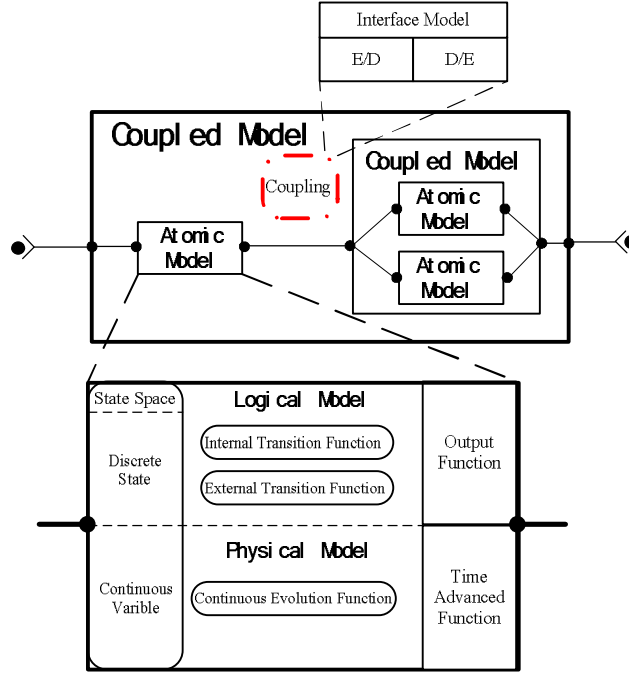


Fig. 2. Structure from HDEVS Formalism

Definition 1. Hybrid atomic models (HAMs)

$$HAM = \langle X, Y, S, L, V, F, d_{int}, d_{ext}, I, ta \rangle \quad (1)$$

- $X = X_{ev} \mathbf{U} X_{si}$: Input; X_{ev} : Set of Discrete Events Inputs; X_{si} : Set of Continuous Signs Inputs
- $Y = Y_{ev} \mathbf{U} Y_{si}$: Output; Y_{ev} : Set of Discrete Events Outputs; Y_{si} : Set of Continuous Signs Outputs
- $S = L \times V$: State Space; L : Set of Discrete States; v : Set of Continuous State Variables
- $F : L \rightarrow (V \times X_{si} \rightarrow V)$, Continuous Evolution Function
- $d_{int} : (L \rightarrow L) \times F$, Internal Transition Function
- $d_{ext} : (Q \times X_{ev}) \times F \rightarrow S$, External Transition Function; $Q = \{(l, v, t_e) \mid t_e \in [0, ta(l)]\}$
- $I : S \rightarrow Y$, Output Function
- $ta : L \rightarrow \mathbb{R}^+$, Time Advance Function

Definition 2. D/E interface models (DEIMs)

$$DEIM = \langle X_{si}, Y_{ev}, T_{ev} \rangle \quad (2)$$

$$T_{ev} : \Omega \rightarrow 2^{Y_{ev}} \quad (3)$$

- $\Omega \in (X_{si}, T)$: Set of X_{si} in Interval T

Definition 3. E/D interface models (EDIMs)

$$EDIM = \langle X_{ev}, Y_{si}, T_{si} \rangle \quad (4)$$

$$T_{si} : X_{ev} \rightarrow Y_{si} \quad (5)$$

Definition 4. Hybrid coupled models (HCMs)

$$HDM = \langle X, Y, D, \{M_i\}, C_{xx}, C_{yx}, C_{yy}, Select \rangle \quad (6)$$

-- D : Name Set of Sub-models

$$\{M_i\} = \{HAM_i\} \cup \{HCM_i\} \cup \{DEIM_i\} \cup \{EDIM_i\} \quad (7)$$

$$C_{xx} = X_{ev} \times \bigcup_i dX_i \cup X_{si} \times \bigcup_j cX_j \quad (8)$$

$$C_{yx} = Y_{ev} \times \bigcup_i dX_i \cup Y_{si} \times \bigcup_j cX_j \quad (9)$$

$$C_{yy} = \bigcup_i dY_i \times Y_{ev} \cup \bigcup_j cY_j \times Y_{si} \quad (10)$$

-- C_{xx} : External Input Coupling (EIC); C_{xy} : Internal Coupling (IC); C_{yy} : External Output Coupling (EOC); dX_i : Set of Input Event in $\{M_i\}$; cX_i : Set of Input Sign in $\{M_i\}$; dY_i : Set of Output Event in $\{M_i\}$; cY_i : Set of Output Sign in $\{M_i\}$

-- $Select : 2^D \rightarrow D$: Tie-breaking Function

In summary, HDEVs formalism divides the combat entity model into two parts and the grouping enables us to reconfigure the overall combat entity model by sharing the same interface within the group. Also, the proposed model partition facilitates concurrent modeling between the two vertical layers at the model development phase. This reduces the complexity of model development and increases the productivity of modeling concepts. For simulation of various engagement scenarios (i.e., reflection of alternative tactical rules or application of diverse search algorithms), modification of the PM layer occurs more frequently than in the case of the LM layer. Therefore, this model method minimizes revision of models without modifying LM layers.

UWSO Ontology. Ontologies are conceptual models that capture and explain the vocabulary used in applications guaranteeing communication free of ambiguities in formats [3]. Ideally an ontology should focus on the description of simulation component to facilitate model composition, as well as to support model discovery and reuse within sets of classes. UWSO (Underwater Warfare Simulation Ontology) is a proposed ontology to represent corresponding domain knowledge and detailed models.

As shown in Fig. 3, the structure of proposed UWSO spans two directions. Classes for keywords, parent, child and category are defined to demonstrate the position of entity in the organization of overall model repository. Others represent the individual information of the entity mostly of behavior that is modeled as a finite state machine starting from an initial state and proceeding to a final state based on HDEVs. Transitions are triggered by an arrival event or an elapse in a time interval and change the state, which may produce certain output. After the individual entity is built, the whole models in this domain are organized in the ontology developed in Protégé 4.3 referring to overall model structure of underwater warfare.

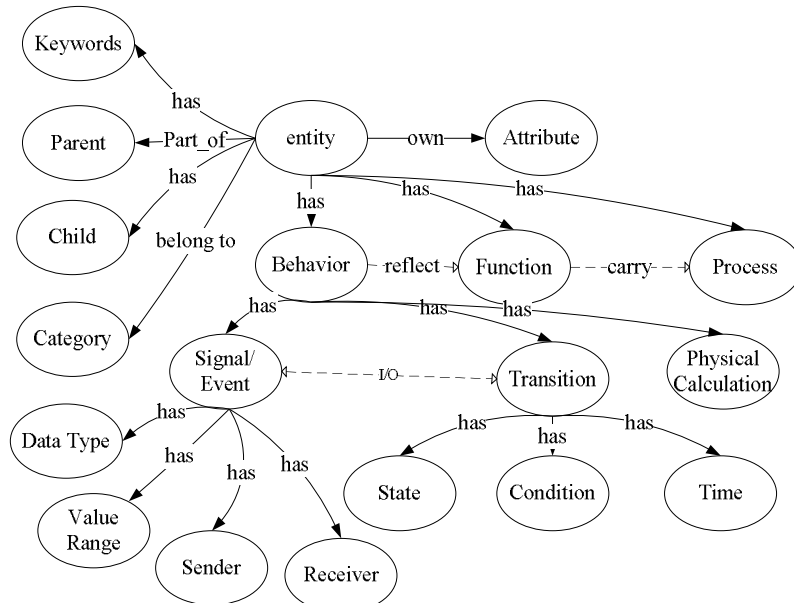


Fig. 3. UWSO Structure

Rule-driven Composition Method

The typical engagement of underwater warfare involves the attacking platform, which fires weapons, and the target platform. Figure 1 shows the partial simulation scenario and its tasks that the platform should accomplish in the engagement stage. Once the attacking platform detects some threats, it keeps in contact with the threat to identify whether it can be a target or not. If the threat is regarded as a target, the platform tracks the target through target motion analysis (TMA) and approaches it to launch weapons under certain conditions. When the target is within the scope, the platform fires and guides its weapons, and finally, the platform makes a detour operation. Consequently, tasks for the attacking platform are relevant to patrol, target detection/identification/tracking, attack, tactical evasion and so on. To be concrete, an attacking platform can be represented by a single or integrated component model which has several behavioral states, and the states represent certain tasks. Additionally, conversion of tasks that implies engagement flow is described as a state transition in the model, and a task is performed in the corresponding state. Therefore, the simulation scenario can be viewed as a discrete event simulation exchanging signals and interactions between the models, and this is why we have widely used discrete event modeling and simulation in the military domain system.

The composition process comprises two phases: model search; model matching and composition. During the simulation development process, the simulation scenario is parsed into sets of tasks. Then we start discovery, matching and composition (DMC) process that consists of the follow steps.

- 1) The keyword-based query is built based on first task and sent to the model repository. The repository returns a set of potential candidates corresponding to the query.
- 2) The candidate models are matched syntactically, semantically and pragmatically with the designed rules.
- 3) Search models about the next task, the result including candidate models can be used as refined condition for the final result.

Iteratively execute the steps above until all tasks have been done and the composition rules (See Tab. 1) are depicted as follows.

In the syntactic layer, composition is operated by checking syntactic aspects of actions and messages that are concerned with the compatibility of implementation details.

(Rule-1): The name of the message including types of event and signal in the simulation scenario with the one in the candidate model should be exactly matched.

(Rule-2): The mode of action associated with the message should be a combination of “In” action of a model with respective “Out” action of other models accepting the message.

(Rule-3): The quantity of parameters of an event or signal in the model with those of the candidate should be equal.

In the semantic layer, composition is operated by checking semantics of the message by using the semantic information. This semantic information consists of data type and unit of each parameter as well as the ontology of the event initiator and receiver entities.

(Rule-4): The entity type for message sender and the one which receives the message in both models should accord with domain knowledge of corresponding model hierarchy, such as the event of launching torpedo should relate with a platform and a torpedo, or a weapon module belong to certain platform and a torpedo.

(Rule-5): The unit of a parameter in both sides should be the same or can be convertible without loss of information, such as conversion between mile and kilometer.

(Rule-6): The type of parameters in both sides should be compatible.

(Rule-7): The range of output parameters in message sender should be in the scope of input parameters of the model that receives the message.

In the pragmatic layer, composition is operated by checking the “pre/post conditions” defined as the relation between post-condition of an “Out” action and the pre-condition of the corresponding “In” action.

(Rule-8): The implication ($\text{PreCond_Send} \wedge \text{PostCond_Send} \rightarrow \text{PreCond_Receive}$) should be held between message sender and the corresponding message receiver.

Tab 1. Composition Rules

Composition Layer	Element to be matched	Content	Rule
Syntactic	Message	Event/Signal Name	Rule-1
	Action	Mode	Rule-2
	Parameters	Number of Parameters	Rule-3
Semantic	Entity	Type	Rule-4
	Parameters	Unit	Rule-5
		Data Type	Rule-6
		Value Range	Rule-7
Pragmatic	State Machine	Pre/Post Condition	Rule-8

Case Study

A typical underwater warfare scenario: one-to-one engagement, a submarine attacks a surface ship. Once the submarine detects the warship by sonar, it keeps in contact to identify. When target is confirmed as enemy, the submarine tracks the target through target motion analysis (TMA) and launches a torpedo. After the submarine fires, it guides the torpedo or let the torpedo autonomous moves. If the warship finds approaching torpedo, it makes a detour to be far from the torpedo. Relevant simulation models are implemented in ADEVS which is a library written in C++ language and provides simulation engine for discrete systems [9]. And many algorithms are realized in form of dynamically linked library (DLL) and embedded into the functions.

In this scenario several key tasks are involved with the submarine, such as patrol, detection, attack. Take the action of launching a torpedo as example, the sender and receiver are submarine and torpedo. First, we discover the needed entities in the model repository through the keyword-based query. The keyword includes name of entities (submarine, fire control and torpedo), types of entities (platform, underwater equipment and underwater weapon) and names of functions (launch and attack). After discover phase, a candidate set is produced. Next is the matching and composition phase. In syntax layer, the models of submarine and torpedo must have the interface of “launch torpedo” and there should be five parameters (time, position, sound level, velocity and heading) in the interface. When in static semantic matching, the torpedo should belong to the class of submarine torpedo, such as it support the function of wire-guide. The unit, data type and value should be correct or convertible. Usually the torpedo adopts knot to show its speed and the data always belongs to type of double to meet the user requirement. In action of launching (Fig. 4), the pre-condition is that the target is detected successfully and post-condition is that the state of torpedo is startup. After finding the target, the submarine keeps in “weapon launch” state and sends the event of launching torpedo to the torpedo fire control system. As a result, the torpedo receives the event and goes into “starting” state. At the same time, the submarine starts to guide the torpedo. Here the conditions are also checked and the “In” and “Out” actions are matched both syntactically and semantically.

After the accomplishment of all tasks involved in the scenario, we obtain the overall execution sequence of model that is input to the simulation engine. The simulation result (Fig. 5) clearly demonstrates the process of combat and indicates the feasibility of proposed method.

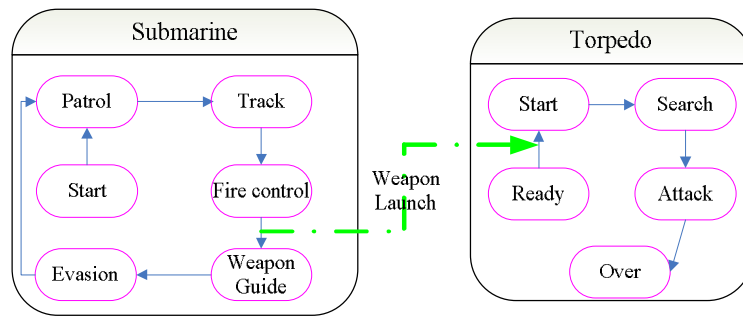


Fig. 4. Information of State Machine Associated with Launch Event

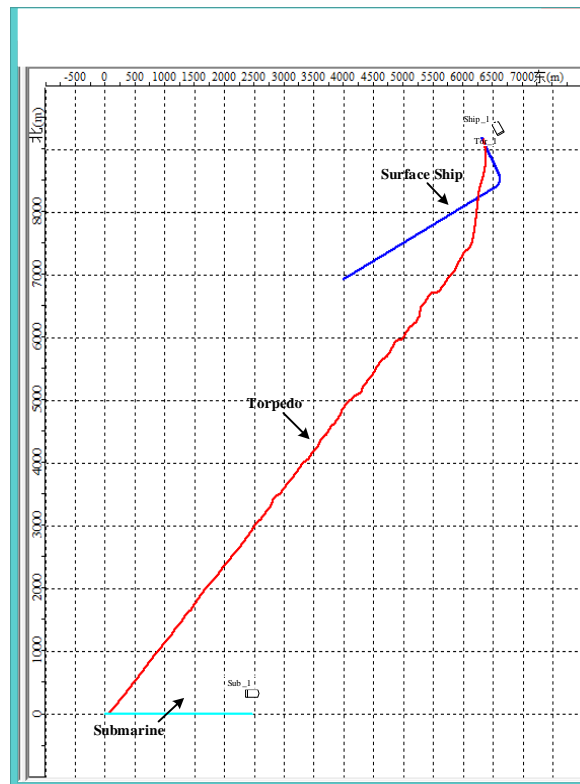


Fig. 5. Simulation Result

Conclusion

In this paper, a structural modeling approach and a rule-driven model composition method in syntactic, semantic and pragmatic layers for underwater warfare simulation are proposed to facilitate the reusability and composability. The combination technology of SES, HDEVS and ontology promotes the reuse and share of various models. And the composition method improves the efficiency of model integration.

References

- [1] Andreas Tolk, Engineering Principles of Combat Modeling and Distributed Simulation, Wiley, 2012.
- [2] C. H. Sung, and T. G. Kim. Framework for Simulation of Hybrid Systems: Interoperation of Discrete Event and Continuous Simulators Using HLA/RTI, 25th ACM/IEEE/SCS Workshop on PADS, (2011) 1-8.
- [3] N. Guarino, D. Oberle and S. Staab. What is an ontology? Handbook on Ontologies. Springer, Berlin, 2009.
- [4] E. W. Weisel, M. D. Petty, and R. R. Mielke, Validity of Models and Classes of Models in Semantic Composability, in Proceedings of the Fall 2003 SIW, Orlando FL, 2003.

- [5] M. D. Petty, E. W. Weisel, R. R. Mielke, Overview of a Theory of Composability, Virginia Modeling Analysis & Simulation Center, Old Dominion University, 2004.
- [6] Y. Hu, G. Tan, F. Moradi, Automatic SOM Compatibility Check and FOM Development, in Proceedings of 7th IEEE Distributed Simulation and Realtime Applications, The Netherlands, 2003.
- [7] C. Szabo, Y.M. Teo, On Syntactic Composability and Model Reuse, in Proceedings of the first Asia Modelling Symposium, 2007.
- [8] <http://en.wikipedia.org/wiki/DEVS>
- [9] <http://web.ornl.gov/~lqn/adevs>