Towards an Open Simulation Scenario Infrastructure

Bikash C. Karmokar[1*], Bharvi Chhaya[2],

Umut Durak[3,4], Shafagh Jafer[2], Sven Hartmann[4]

[1]Leibniz Universität Hannover, Welfengarten 1, Hannover, 30167, Germany;
[*]*bikash.chandra.karmokar@stud.uni-hannover.de*

[2]Embry-Riddle Aeronautical University, 600 S Clyde Morris Blvd, Daytona Beach, Florida 32114, United States
[3]German Aerospace Center (DLR), Institute of Flight Systems, Braunschweig, 38108, Germany
[4]Clausthal University of Technology, Clausthal-Zellerfeld, 38678, Germany

Abstract. Scenarios are crucial elements in a simulation study. Typical scenario management activities include scenario development, loading and parsing, distribution and role casting, event injection, and finally data collection and logging. Recently, significant effort is being spent towards standardization in simulation scenario development. A standard simulation scenario definition language requires an underlying metamodel. This paper presents a System Entity Structure (SES) based metamodeling approach for achieving a standard scenario definition language as a significant step towards open scenario infrastructure.

1 Introduction

A model is a formal specification of a concept, and a simulation is a specific implementation or execution of a model [1]. The execution of any simulation requires a clearly-defined scenario. A simulation scenario is usually defined as the specification of a set of properties, which are (1) initial and terminal conditions, (2) significant events and their time line, (3) the environment, as well as (4) the major entities, their capabilities, behavior and interactions over time.

Scenarios are important artifacts in the simulation engineering process [2]. They not only define specification of a simulation run, but also provide an input for the design and evaluation of the simulation environment itself [3].

Scenario management is required due to the extensive need for scenarios in simulation. Scenario management includes all activities from the development to the execution of the scenario [4]. These are classified into two categories, the activities that take place during (1) design and development of the scenario, and (2) the execution of the scenario [5].

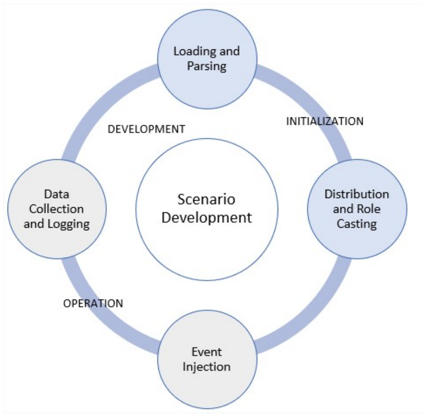

Figure 1: Scenario management activities.

Figure 1 shows the activities associated with scenario management. Scenario development is the only major activity included in the design and development category. During execution, the various activities involved are (1) scenario loading and parsing, (2) scenario distribution and role parsing, (3) event injection, and (4) scenario-related data collection and logging [5].

Scenario management is currently an extensive process due to the lack of common understanding, standardized practice and common tools. The need for standardization and an open infrastructure for scenario management is apparent. There are recent studies in aviation [6, 7] and in automative [8] towards standardization in simulation scenario development. This paper presents a System Entity Structure (SES) based meta-

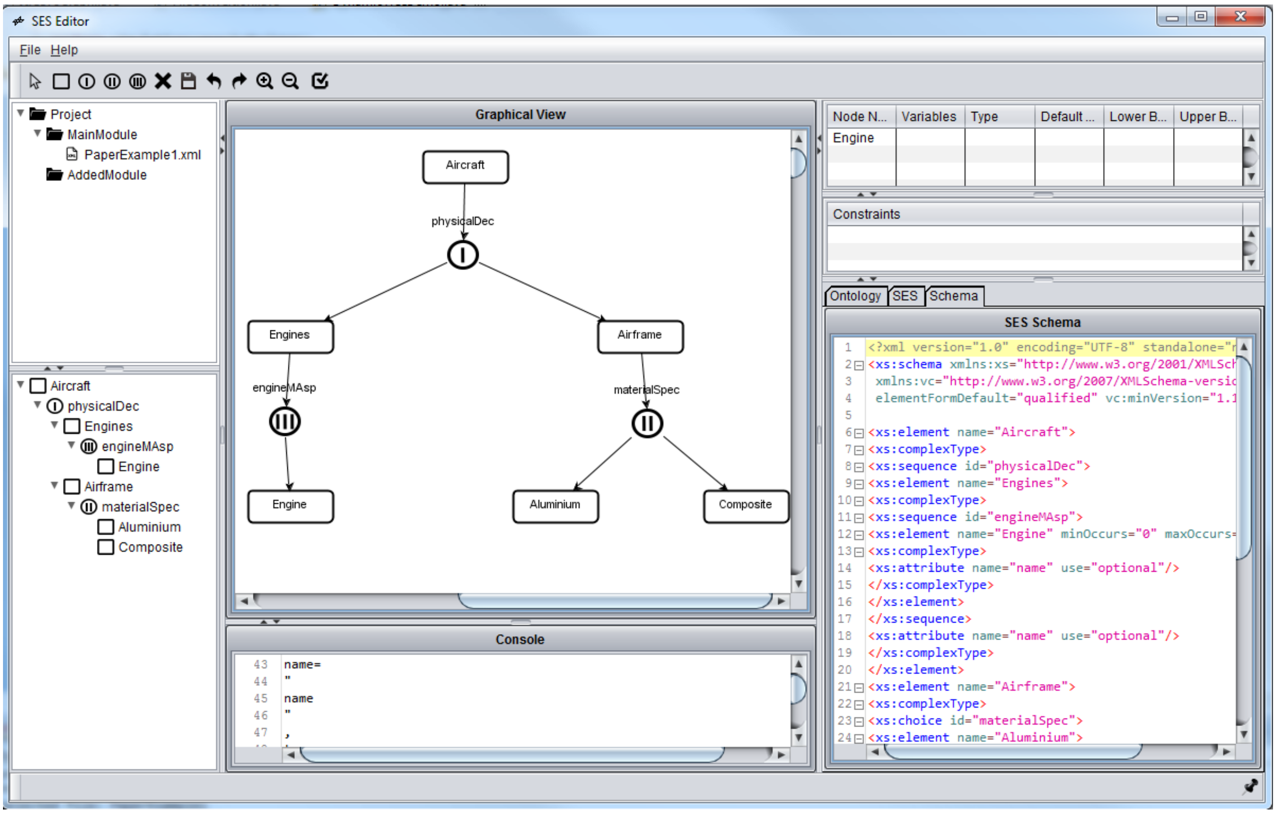

Figure 2: SES Meta-model in SESEditor.

modeling approach and supporting tooling for achieving a standard scenario definition language. The approach has first been proposed by Durak et.al [9] as a significant step towards open scenario infrastructure. The tooling is still under development.

2 Scenario Development

Scenario development is viewed as the transformation of operational scenarios (defined using natural language) to conceptual scenarios (conforming to a metamodel), and eventually to executable scenarios (specified using a scenario specification language). As the development process occurs, the scenario models are refined and transformed [10, 3]. Scenario development starts with operational scenarios, mostly in text form constructed in natural language by domain experts. They usually present the key information from the user's perspective and are often not complete and consistent. Operational scenarios are then used to develop conceptual scenarios, essentially to be based on a metamodel, either an explicit or an implicit one. The result is then a complete and a consistent scenario specification, often compiled in a table. Finally conceptual scenarios are transformed into executable scenarios for target simulation environments using a set of rules specific to the target simulation environment.

3 Standardization in Scenario Development

In order to standardize the scenario development process, we propose standard metamodels which can be used as a basis for creating conceptual scenarios.

Recently, an effort has begun to develop a formal scenario definition language for aviation [11]. The Aviation Scenario Definition Language (ASDL)[6] is a domain-specific language (DSL) which aims to provide a simple, standardized method for aviation simulation scenario generation. ASDL aims to provide a common mechanism for verifying and executing aviation scenarios, allow for effective sharing of scenarios among various simulation environments, improve the consistency of different simulators and simulations, and enable the

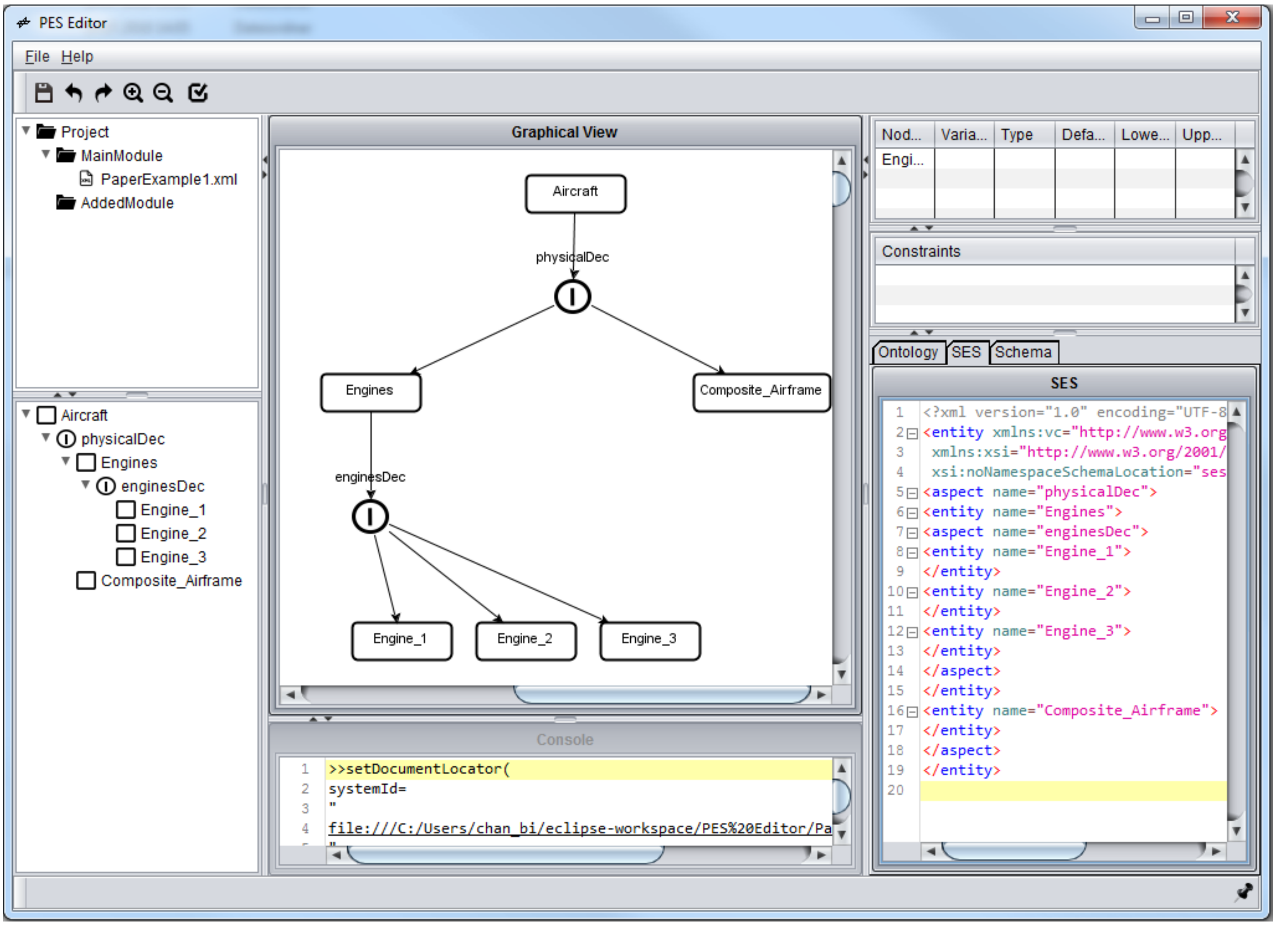

Figure 3: Pruning example in PESEditor.

reuse of scenario specifications.

In the automotive industry, OpenSCENARIO aims to establish a software-independent format of describing dynamic scenarios for virtual driving tests [12, 8]. This standardized scenario format would have potential applications in a wide range of automotive uses such as driver assistance, autonomous driving, test of sensors and algorithms and simulating driving dynamics.

4 System Entity Structure

Conceptual scenarios can be defined and described either graphically or textually using a scenario definition language. Scenario definition language can be introduced as a Domain Specific Languge (DSL). Metamodels are used to define DSLs. One such method of describing metamodels is System Entity Structure [13]. SES is a high-level ontology which was introduced for knowledge representation of decomposition, taxonomy and coupling of systems [14]. SES has a limited set of elements and axioms which makes it an accessible metamodeling approach.

There are four elements of SES [15], which are (1) Entity, (2) Aspect, (3) Specialization and (4) Multiple-Aspect. Entity is an object of interest, and can also have variables attached to it. An Aspect is used to denote a decomposition relationship. The taxonomy of an Entity can be represented by using Specialization. Multi-Aspect is a special kind of Aspect, which represents a multiplicity relationship that specifies the parent Entity as a composition of multiple Entities of the same type.

There are six axioms of SES [16], which are (1) uniformity, (2) strict hierarchy, (3) alternating mode, (4) valid brothers, (5) attached variables, and (6) inheritance. Uniformity confirms that any two nodes with the same label have isomorphic subtrees. Strict hierarchy is used to prevent a label from appearing more than once down any path of the tree. Alternating mode recommends that if a node is an Entity, then the successor is either Aspect or Specialization, and vice versa. The valid brothers axiom disallows two brothers from having the same label. Attached variables specifies a constraint that variable types attached to the same item shall have distinct names. According to inheritance, it is indicated that a Specialization inherits all variables and Aspects.

Pruning is described as resolving the choices in Aspect, Multi-Aspect and Specialization relations and assigning values to the variables. It also specifies the cardinalities in Multi-Aspect relations. The outcome of pruning is called Pruned Entity Structure (PES), which is a selection-free tree.

Metamodels can be developed using SES following the approach proposed in [9]. The resulting Scenario SES captures all scenario elements and their relationships. SES specifies the simulation scenario definition language, and its pruning yield a Pruned Entity Structure (PES), which represents a particular scenario. The computational representation of SES and PES for scenario development has been proposed in [17]

5 Using SES Editor for Metamodeling

In SESEditor, SES is represented by a directed tree structure. Here, objects are represented by nodes which are connected using edges. Figure 2 shows an SES model created in SESEditor, where the root entity Aircraft is decomposed into Engines and Airframe using physicalDec aspect node. MultiAspect node enginesMAsp decomposes Engines to Engine, representing that Engines is made of multiple instances of Engine. Specialization node materialSpec is used to express that Airframe can be Aluminum or Composite.

6 Using PES Editor for Pruning

Figure 3 shows a PES of Aircraft Metamodel in PESEditor. During pruning, MultiAspect node enginesMAsp is pruned and aspect node enginesDec is added with children Engine_1, Engine_2 and Engine_3 of the same type as Engine. From the available options, specialization node is pruned and Composite_Airframe is added. Thus the completely pruned structure is created where there are no choices left.

7 Conclusions

The scenario management process consists of scenario development and several activities during scenario execution. A brief overview of the current effort towards standardizing scenario development in the aviation and automotive industries was presented here. The paper highlighted the use of System Entity Structure (SES) and its pruned counterpart PES for metamodeling scenario specification languages and for creating specific scenarios from it. The use of SESEditor and PESEditor was demonstrated to display a small metamodel of an aircraft entity. These editors allow graphical representation of metamodels and pruning for effective scenario generation.

References

[1] Sargent RG. Verification and validation of simulation models. In: *Simulation Conference (WSC), Proceedings of the 2009 Winter*. IEEE. 2009; pp. 162–176.

[2] IEEE. IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP). *IEEE Std 1730-2010*. 2011;.

[3] Durak U, Topçu O, Siegfried R, Oguztuzun H. Scenario development: A model-driven engineering perspective. In: *2014 International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*. IEEE. 2014; pp. 117–124.

[4] Löfstrand B, Ericsson M, Johansson M, Strand J, Lepp H. Scenario management-Common design principles and data interchange formats. In: *European Simulation Interoperability Workshop (SIW), 04E-SIW-070*, vol. 46. 2004; .

[5] Topçu O, Durak U, Oğuztüzün H, Yilmaz L. *Distributed simulation: A model driven engineering approach*. Springer. 2016.

[6] Jafer S, Chhaya B, Durak U, Gerlach T. Formal scenario definition language for aviation: aircraft landing case study. In: *AIAA Modeling and Simulation Technologies Conference*. 2016; p. 3521.

[7] Durak U, Jafer S, Beard SD, Reardon S, Murphy JR, Crider DA, Gerretsen A, Lenz H, Macchiarella ND, Rigby KT, et al. Towards a Standardization for Simulation Scenario Development in Aviation-Panel Discussion. In: *2018 AIAA Modeling and Simulation Technologies Conference*. 2018; p. 1395.

[8] OpenSCENARIO bringing content to the road. www.openscenario.org/. Accessed: 2018-07-25.

[9] Durak U, Pruter I, Gerlach T, Jafer S, Pawletta T, Hartmann S. Using System Entity Structures to model the elements of a scenario in a research flight simulator. In: *AIAA Modeling and Simulation Technologies Conference*. 2017; p. 1076.

[10] Jafer S, Chhaya B, Updegrove J, Durak U. Schema-based Ontological Representations of a Domain-Specific Scenario Modeling Language. *Journal of Simulation Engineering*. 2018;1.

[11] Reardon SE, Beard SD, Lewis E. Scenario Development at the Vertical Motion Simulator. 2017;.

[12] Barsi A, Poto V, Tihanyi V. Creating OpenCRG Road Surface Model from Terrestrial Laser Scanning Data for Autonomous Vehicles. In: *Vehicle and Automotive Engineering*. Springer. 2018; pp. 361–369.

[13] Durak U, Pawletta T, Oguztuzun H, Zeigler BP. System entity structure and model base framework in model based engineering of simulations for technical systems. In: *Proceedings of the Symposium on Model-driven Approaches for Simulation Engineering*. Society for Computer Simulation International. 2017; p. 1.

[14] Kim TG, Lee C, Christensen ER, Zeigler BP. System entity structuring and model base management. *IEEE Transactions on Systems Man and Cybernetics*. 1990; 20(5):1013–1024.

[15] Zeigler BP, Hammonds PE. *Modeling and simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange*. Elsevier. 2007.

[16] Zeigler BP. Multifacetted modelling and discrete event simulation. 1984;.

[17] Durak U, Jafer S, Wittman R, Mittal S, Hartmann S, Zeigler BP. Computational Representation for a Simulation Scenario Definition Language. In: *AIAA SciTech Forum*. 2018; .