Ontology for Objective Flight Simulator Fidelity Evaluation

Umut Durak[1*], Artur Schmidt[2], Thorsten Pawletta[2]

[1] German Aerospace Center (DLR), Institute of Flight Systems, Lilienthalplatz 7, 38100 Braunschweig, Germany; *umut.durak@dlr.de

[2] University of Wismar, FIW / MVU, FG CEA, PF 1210, 23952, Wismar, Germany

Simulation Notes Europe SNE 24(2), 2014, 69 - 78
DOI: 10.11128/sne.24.tn.10242
Submitted: Sept. 15, 2014 (selected ASIM SST Post-Conf. Publ.);
Revised: Oct. 20, 2014; Accepted: October 30, 2014;

Abstract. The term simulator fidelity has become enormously important in the scope of simulation research, when assessing training efficiency and the transfer of training to real flight. It is defined as the degree to which a flight simulator matches the characteristics of the real aircraft. Objective simulator fidelity provides an engineering standard, by attacking the fidelity problem with comparison of simulator and the actual flight over some quantitative measures. Research flight simulators encompass some differences from commercial flight simulators. They require high flexibility and versatility concerning the cockpit layout and visual and motion systems, as well as flight simulation models. It should be easy to modify the flight simulation model or other software and hardware components of the simulator. To support this, there is a need for a flexible automated test methodology, in order to determine the fidelity of the most relevant simulator subsystems, since they are often modified during the life cycle of the simulator. This methodology not only shall support automated execution but also enable automated generation of the test cases which are subject to change as well as simulator components. The Institute of Flight Systems (FT) at the German Aerospace Center (DLR) has a reconfigurable flight simulator, the Air Vehicle Simulator (AVES), for research of rotorcraft and fixed-wing aircraft.

The study reported in this paper adopts a Model Based Testing approach to tackle the high flexibility requirement of AVES. The outcome of the paper is a metamodel for model-based objective flight simulator evaluation. Metamodeling has been carried out in two levels. An Experimental Frame Ontology (EFO) has been developed adopting experimental frames from Discrete Event System Specification (DEVS), and as an upper ontology to specify a formal structure for a simulation test. Then in Objective Fidelity Evaluation Ontology (OFEO) that builds upon EFO, domain specific meta-test definitions are captured.

Introduction

Since the late 1920s, when Edward Link built the 'Blue Box' [1], flight simulators have been important elements of aviation. Flight simulators became well accepted as training aids by many aircraft operators before the digital era. Highly sophisticated flight simulators have been employed commercially within civil and military flight training organizations in order to enhance pilot skills.

In the 1980s, the aeronautics research community started using flight simulators for developing and experimenting advanced concepts and conducting aviation human factors research. Some of the first examples of research flight simulators include ATTAS Ground-Based Simulator from German Aerospace Center (DLR) [2] [3], National Aerospace Agency (NASA) Crew Vehicle Systems Research Facility in Ames Research Center [4] and Visual Motion Simulation and Cockpit Motion Facility at the Langley Research Center [5]. Some more recent examples are the Air Vehicle Simulator (AVES) of DLR [6], HELIFLIGHT at the University of Liverpool [7], NASA Ames Vertical Motion Simulator (VMS) [8] and International Research Institute for Simulation, Motion and Navigation (SIMONA) of Delft University of Technology [9].

The authors define fidelity in flight simulation as the degree to which a flight simulator matches the characteristics of the real aircraft. As its effect on training efficiency and transfer of training to real flight became better understood, fidelity became a more important research subject [10]. Objective simulator fidelity assessment provides an engineering standard to qualify the degree of fidelity through objective measures. It approaches the fidelity problem with comparison of simulator and the actual flight over some quantitative cues.

Requirements for research flight simulators encompass some differences from commercial flight simulators.

They require high flexibility and versatility concerning the cockpit layout and visual and motion systems, as well as flight simulation models. They must allow easy modification of the flight simulation model or other software and hardware components of the simulator. In order to efficiently determine the fidelity of subsystems that are often modified during the life cycle of the simulator, there is a need for a flexible automated test methodology.

This methodology is required to automate not only the execution, but also the test case generation. While there are standard sets of test cases for objective flight simulator evaluation, each modification of simulator components asks for either a different subset of a standard test set or modifications in standard test specifications. Therefore, test cases are also required to be easily modifiable, as well as the components of a research simulator.

Automated testing can be applied through the use of software to control the execution of tests and a comparison of actual outcomes to the predicted ones. Available test data taken from aircraft are used as input signals to the simulator and the output signals of the simulator are compared to the measurements to be presented for the evaluator in a smart format. Braun and Galloway [11] reported their automated fidelity test system that compares directly the flight test results and manual execution of flight tests in simulators.

Wang et al. [12] [13] presented *Automated Test System* (ATS) that measure force function, evaluation function and transport delay with its non-intrusive interface with operator station. Jarvis et al. [14] summarizes the efforts on validation of sensory cues, motion cues, vibration and sound cues, visual cues, transport delays and flight dynamics models in flight simulators.

Previous efforts regarding automated testing for objective flight simulator evaluation utilized fixed test descriptions. The presented automated testing infrastructures contributed flawless execution of the tests. But they did not attack automation of test case generation. The bridge between the state of the art Model Based Testing (MBT) practices and automated flight simulator testing is still missing. MBT can be introduced as the idea of automating test case generation from a test model rather than implementing test cases manually [15].

Thus, the test case generation is made more flexible. Metamodeling is employed to capture the domain specific concepts and constraints for building test models.

Then test modeling is used to specify test cases, and these test models are translated automatically to executable test cases [16]. DLR intends to adopt an MBT approach in flight simulator domain and hereby provide a methodology for flexible automated test case generation. Therefore a metamodel is required for objective flight simulator fidelity evaluation.

A metamodel is defined as an explicit model of constructs and rules that are used to define a model [17]. Following Gruber [18], definition of ontology is “explicit specification of shared conceptualization”. Moreover, metamodels are categorized as ontologies that are used by modelers [17].

Here, the test case can be defined as a sequence of input stimuli that will be fed to the System Under Test (SUT), namely test inputs and the expected behavior of the system, namely test oracle (**Figure 1**) [19].

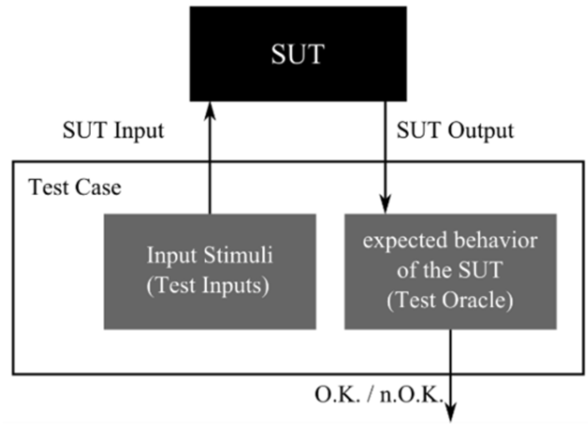

Figure 1. Test Case Structure.

Moser et al. [20] stressed that ontologies as machine-readable domain knowledge, which can be utilized for test case generation. Then Nguyen et al. [21] presented a framework for ontology driven test case generation in the context of multi-agent systems. Adopting these ideas, ontologies are employed to structure meta-test definitions.

The domain knowledge about the objective validation of simulator systems including the rules for assessing the results of test runs is captured in ontologies.

Zeigler and his colleagues developed the concept of *Experimental Frame* (EF) [22] [23]. An EF defines the conditions under which a model is to be examined. It comprises of an input generator, a verifier for the desired conditions and an analyzer for the outputs.

Following Zeigler et al. [23], the EF is critical for evaluating the model validity. Traoure and Muzy in [24] and Foures et al. in [25] published the usage of the EF approach for specifying invariant validation experiments.

In this research, metamodeling has been carried out on two levels. An *Experimental Frame Ontology* (EFO) has been developed as an upper ontology to specify a formal structure for generic simulation test model. Then in *Objective Fidelity Evaluation Ontology* (OFEO) that builds upon EFO, domain specific meta test definitions are captured. Protégé [26] is used as the ontology development environment and ontologies are developed using *Ontology Web Language* (OWL).

This paper will present these ontologies after introducing a background on objective fidelity evaluation, experimental frames and ontologies in general.

In this paper, first a background will be introduced on objective fidelity evaluation, EF and ontologies. Then EFO and OFEO will be presented. The paper will end with concluding remarks.

1 Background

1.1 Objective fidelity evaluation

Fidelity is regarded as a multivariate construct with no consensus among researchers on a single index of measurement or definition and it is strongly related to the training task to be performed with the simulator.

There are two approaches to measure simulator fidelity; the subjective and objective approaches [12]. The subjective approach tries to identify the degree of realism felt by the user. User feedback is usually collected using subjective rating scales [27].

Although subjective scales are valuable, it is hard to generalize across scales because of the individual opinions and bias of those providing assessments [12]. Objective approaches attack the fidelity problem with of simulator and the actual flight over some quantitative cues.

'ICAO 9625 Manual of Criteria for the Qualification of Flight Training Devices' [28] is the well accepted global standard for qualification of flight training devices. The standard specifies seven types of fidelity that correspond to a capability level to provide a certain type of training.

For example, simulators classed as 'Type 1' can be used for all training tasks used during completion of Private Pilot License (PPL) training, whereas 'Type 7' is required for some of the training tasks used when awarding 'Type Rating'. Appendix B of the standard specifies the test cases for objective validation of simulators. These test cases include comparison of results from tests conducted in the simulator and aircraft validation data.

The Royal Aeronautical Society (RAeS) published 'Aeroplane Simulation Training Device Evaluation Handbook Vol. 1 Objective Testing' [29] to ease the implementation and enhance the understanding of objective tests introduced in ICAO 9625. It provides further discussions about the implementation of each test and introduces some example cases with some plots. ICAO 9625 provides tables that specify each test case with parameters, tolerances and flight conditions. Table 1 shows an example test specification from the standard, for testing the minimum radius.

Test	Tolerance	Type						
		1	2	3	4	5	6	7
Minimum radius turn	±0,9m (3ft) or ±20% of aeroplane turn radius					✓		✓

Table 1: Sample Test Specification from ICAO 9625 [28]

This effort takes ICAO 9625 as a baseline to define test cases as they present a shared understanding of experts in the field. Tests are grouped under performance, handling qualities, motion system, visual system and sound system. Among these tests, those regarding performance and handling qualities are related to flight dynamics models, and have no other subsystem or device dependencies. For this reason, they are considered to better suit automation. Therefore, as a first step, the current research addresses these groups.

The RAeS introduces the benefits of employing automatic testing in objective fidelity evaluation as repeatability, ease and rapidity of conducting tests. The RAeS handbook [29] specifies the features of an automatic testing system as initializing the simulator with the test initial conditions, trimming the aircraft, creating the stimulus if required, using flight controls and finally checking the simulator output against test criteria.

1.2 Experimental frame approach

The EF approach was originally introduced by Zeigler in [22] in context with the *Discrete Event System Specification* (DEVS). The objective is the explicit separation between the model and the experiment. Moreover, an EF specifies a limited set of circumstances under which a model is to be observed. Currently, the EF approach belongs to the state of the art and it is used in many modelling and simulation projects including validation experiments [24] [25] [30] [31]. Following Zeigler [22], the formal specification of the EF is given by the 7-tuple:

$$\mathrm{EF} = < \mathrm{T}, \mathrm{I}, \mathrm{O}, \mathrm{C}, \Omega\mathrm{i}, \Omega\mathrm{c}, \mathrm{SU} >$$

where:

T is the time base
I is the set of input variables
O is the set of output variables
C is the set of control variables
Ωi is the set of admissible input segments
Ωc is the set of admissible control segment
SU is a set of summary mappings

The EF can be implemented in various ways. Zeigler [22] recommends implementing the EF as a coupled system consisting of a generator, acceptor and a transducer that is connected to a SUT. In our context, the SUT is always a model. For this reason, it is called *Model Under Test* (MUT). Figure 2 illustrates such a realization of EF coupled to a MUT schematically.

Test inputs are produced by a generator. The set of admissible input segments influences MUT's behavior. The acceptor and transducer form the test oracle. Based on output variables, the transducer calculates outcome measures in the form of performance indices, comparative values, statistics etc.

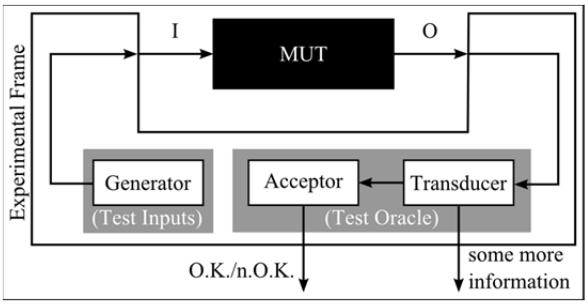

Figure 2: Illustration of EF with MUT.

The acceptor corresponds to a decision unit that decides if an experiment is valid or not. For this purpose, the acceptor monitors its inputs and maps them to a specified admissible control segment. In case of violation of the admissible control segment the experiment will not be accepted. Beside control variables, the input of an acceptor can be output variables or outcome measures.

The EF approach defines a uniform structure for a systematic experiment specification. The specification has to be coded in the description of an EF. This means that each kind of experiment needs the definition of a distinct EF.

1.3 Ontologies

Knowledge in a domain is formalized using concepts, relations, functions, axioms and instances in an ontology. Concepts can be anything about which something is said, and therefore, can be a description of a task, function, action, strategy etc. Taxonomies are widely used to organize the ontological knowledge in domain using generalization/specialization relationship through simple/multiple inheritance.

Relationships represent a type of interaction between the concepts of the domain and functions can be regarded as a special kind of relation. Axioms on the other hand are used to model sentences that are always true. They are added to ontology for several purposes, such as constraining the information contained in the ontology, verifying its correctness or deducting new information. Instances are the terms that are used to represent the elements of the domain. They actually represent the elements of the concepts [32].

Ontologies in engineering domain have been developed for various purposes including specifying engineering information systems, integration of engineering applications, supporting engineering design and development. The first efforts on developing engineering ontologies were in the 1990's. The 'PhysSys' [33] was one of the first engineering ontologies based upon system dynamics theory that is practiced in engineering modeling, simulation and design. The PhysSys was developed to formally define how design engineers or the end users of *Computer Aided Engineering* (CAE) systems understand their domain and to provide a foundation for the conceptual schema for data structuring in engineering databases, libraries and other CAE information systems [33] [34].

The ideas formalized in PhysSys provided a base for the development of a library of reusable models for engineering and design.

Fishwick and Miller in [35] discussed the venues of ontology use in modeling and simulation. One of the late examples of ontology use in modeling and simulation is reported by Durak et al. [36] [37]. The group enabled simulation reuse over an ontology driven methodology.

Another ontology-based modeling and simulation approach was established by Zeigler with the System Entity Structure and Model Base (SES/MB) framework [22] [23] [38] [39].

Today the SES is an ontology framework for conceptual system modeling and for specification of a set of modular hierarchical system structures and parameter settings.

2 Experimental Frame Ontology

The EFO forms the upper level of the metamodel for objective flight simulation evaluation. The previously introduced EF approach is used to specify a formal structure of generic test cases. Hence, every test case has to be specified according to the EF definition in the Section 1.2.

Figure 3 illustrates the entity hierarchy of the EFO in Protégé. The first layer consists of three entities: Computational Unit, Informational Unit and the EF. Computational Units comprises the generic Acceptor, Transducer and Generator which will be presented as executable blocks in a test case. The Information Unit defines basic entities of an EF. The Experimental Frame entity thus conforms to the actual EF.

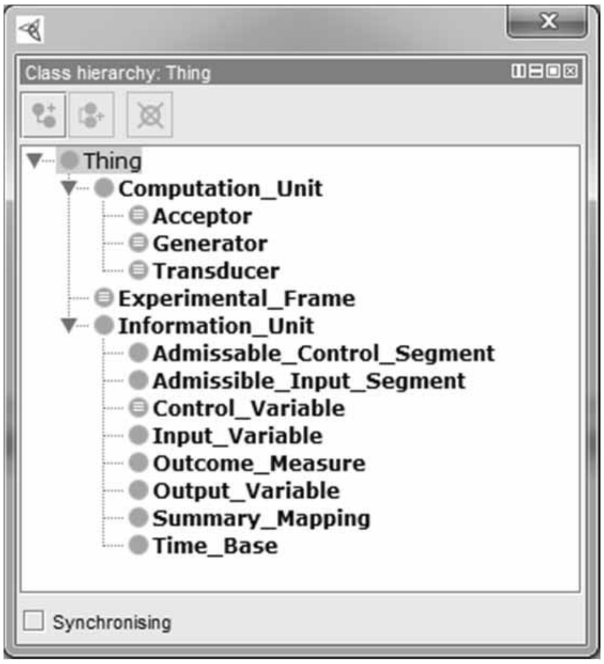

Figure 3: Entity Hierarchy of the Experimental Frame Ontology.

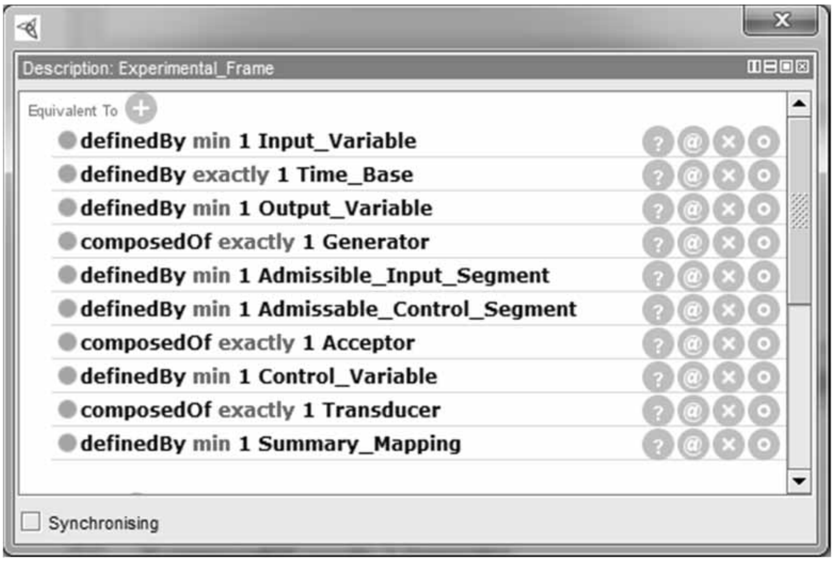

Figure 4: Description of a Generic Experimental Frame.

Furthermore particular properties are implemented to define the relations between the entities. For example the properties *composedOf* and *definedBy* makes clear that any EF is a composition of Computational Unit and is defined by the Informational Units.

As a result we obtain a generic EF which conforms to a generic test case. Thus, any test case will have the unique structure as shown in Figure 4 on its top level. The EFO forms the basis for the OFEO that will define test cases in detail.

3 Objective Fidelity Evaluation Ontology

OFEO is constructed by extending the upper level EFO that specifies any test case that will be applied to MUT using experimental frames formalism. The hierarchy of OFEO using Protégé is depicted in Figure 5. The elements from EFO can be traced in this hierarchy.

Each objective validation test case described in ICAO 9625 under performance and handling qualities are specified by an experimental frame. Thus, each test possesses a Generator, Transducer and an Acceptor. The specification of these three entities will inherently describe how this specific test will be exercised. These three entities will constitute the automatic test system.

Following the features of automated test systems introduced in the RAeS Handbook [29], the Generator is described as the component to initialize the test with initial conditions and trim the aircraft and create the stimulus following the ones from the flight test using the flight controls.

Hence, the Generator is interpreted as test independent. On the other hand, the Transducer is described as the component that will compute Outcome Measures that are required for the Acceptor for a specific test.

As an example, the Minimum Turn Radius test requires a Simulated Turn Radius to be computed from a simulation output. Or likewise, Rate of Turn versus Nosewheel Steering Angle test requires Simulated Turn Rate value to be computed.

So, a specific Transducer is defined for every test. Lastly, the Acceptor is described as the component that checking the MUT against test criteria. Since every test has a particular criterion, an Acceptor is defined for each test. Accordingly, we are expecting to have particular Control Variables for each test.

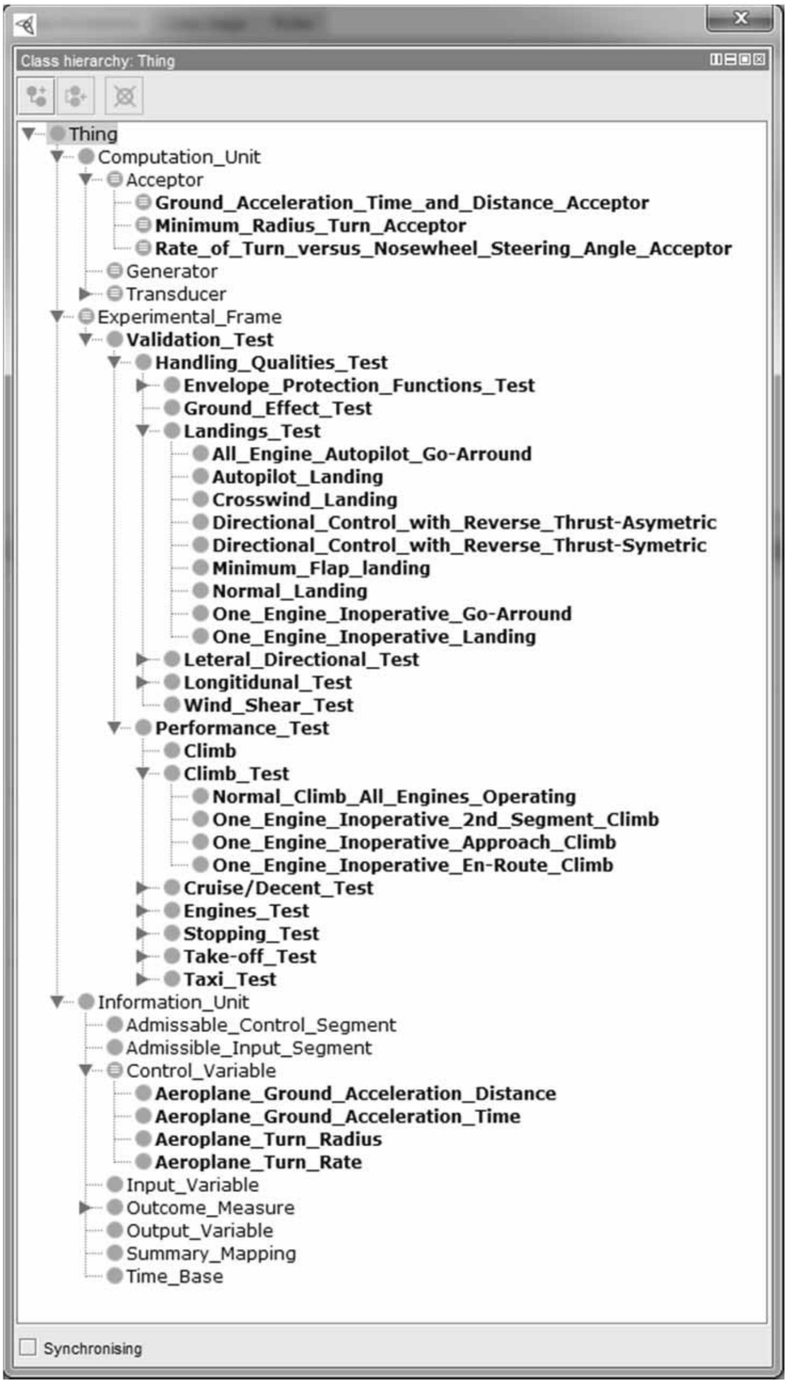

Figure 5: Objective Fidelity Evaluation Ontology Hierarchy.

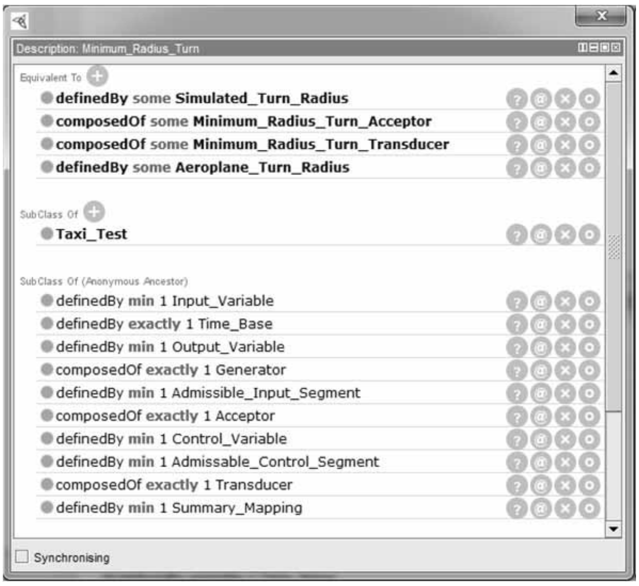

Figure 6: Minimum Radius Turn Test Description.

Figure 6 presents an example test description in Protégé. The Minimum Turn Radius Test is specified with a specific Acceptor, Transducer and Control Variables, namely Simulated Turn Radius and Aeroplane Turn Radius. On the other hand, it inherits the properties of an experimental frame. So it will also have a Generator, Input Variables, Output Variables, Admissible Input Segments, Admissible Control Segments and a Summary Mapping. It is clear that input and output variables of the flight simulator are application specific but does not vary with test cases, so generic definitions are kept for these variables and admissible segments.

Minimum Radius Turn Transducer (Figure 7) is defined with an output Simulated Turn Radius while it also inherits the properties of a Transducer. It will be using Output Variables for computing the outcome measure. Since the computation of the outcome measure is largely implementation specific, ontology does not have any knowledge about it.

As an example, the Minimum Radius Turn Acceptor is depicted in Figure 8. Since each of the tests has distinct criteria, the Acceptors will have particular inputs. Accordingly, Minimum Radius Turn Acceptor is described with Simulated Turn Radius and Aeroplane Turn Radius inputs.

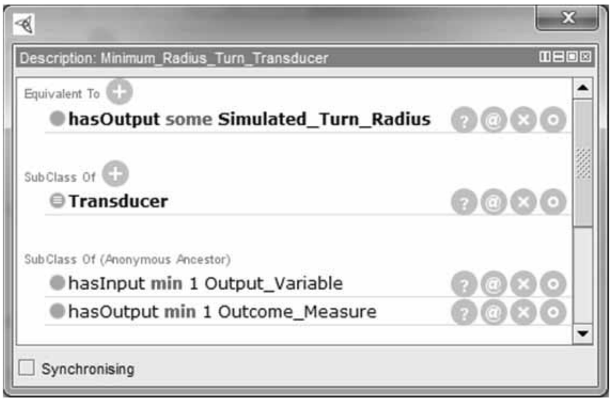

Figure 7: Minimum Radius Turn Transducer Description.

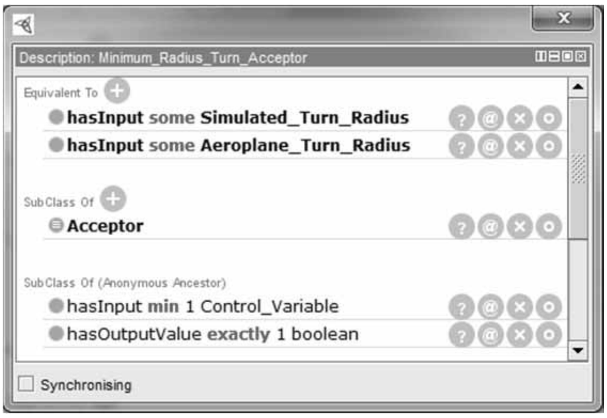

Figure 8: Minimum Radius Turn Acceptor Description.

On the other hand the output of the Acceptor is always a Boolean. It reports if the criterion is matched or not.

Semantic Web Rule Language (SWRL) [40] is used to formalize the acceptance criteria. SWRL can be regarded as an extension to OWL to specify rules for enhancing expressivity.

Thus rule-based reasoning over the knowledge captured in an ontology is possible. In this study, rules specify how the inputs of the Acceptor are used to compute if the test is successful or not. In

Figure **9**, the rule in the front windows says that Minimum Radius Turn Acceptor has a true output when the difference between the simulated and the real minimum turn radius is smaller than 20 %.

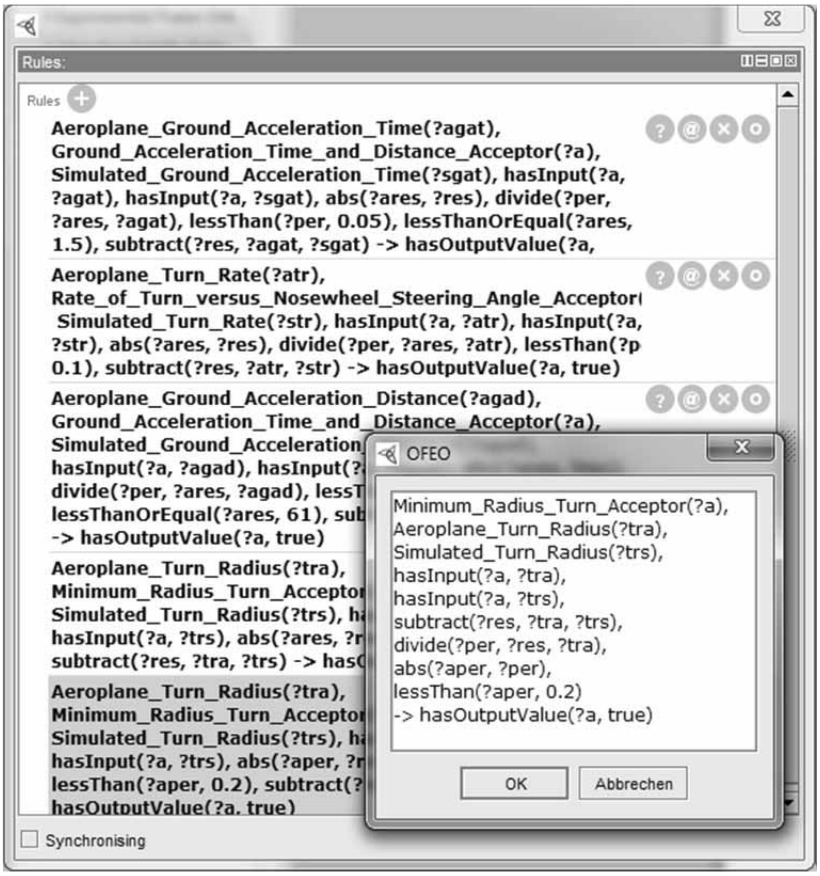

Figure 9: Rules for Acceptors.

While Web Ontology Language is used as the ontology language; Semantic Web Rule Language is employed to capture the rules. Protégé is utilized as the ontology development environment.

This effort assembled the semantic infrastructure for developing model based automated test methodology for simulator fidelity evaluation. The next step is to construct the toolset for developing the test models utilizing the presented metamodels. This toolset set shall also support model transformations to generate executable test cases and execution of these test cases.

Although Web Ontology Language, Semantic Web Rule Language are employed in this metamodeling step, the representation form of the knowledge captured in ontologies may vary in toolset implementation due to practical reasons like platform compatibility.

4 Conclusion

Research simulators require flexible and adoptable test methodologies to accommodate frequent changes to their components. This paper presents an ontology based metamodeling approach for adopting a Model Based Testing methodology for objective flight simulator evaluation.

Experimental Frames Ontology adopts the concept of Experimental Frames from Discrete Event Systems Specification, as an upper ontology to specify a formal structure for test cases.

Thus with Experimental Frames, concepts of Model Based Testing could be formally specified. This established a solid base for modeling specific test cases. Then in Objective Fidelity Evaluation Ontology that builds upon Experimental Frames Ontology, domain specific meta-test definitions are modeled.

References

[1] D. Allerton. *Principles of Flight Simulation.* John Wiley & Sons, Ltd, West Sussex, United Kingdom, 2009.

[2] P. Saager. *Real-Time Hardware-in-the-Loop Simulation for 'ATTAS' and 'ATTHeS' Advanced Technology Flight Test Vehicles*. in AGARD Guidance and Control Panel, 50th Symposium, Izmir, Turkey, 1990.

[3] S. Klaes. *ATTAS Ground Based System Simulator - An Update-*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Denver, CO, 2000.

[4] B. Sullivan and P. Soukup. *The NASA 747-400 Flight Simulator: A Natonal Reseource fir Aviation Safety Research*. In: AIAA Flight Simulation Technologies Conference, San Diego, CA, 1996.

[5] R. Smith. *A Description of the Cockpit Motion Facility and the Research Flight Deck Simulator*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Denver, CO, 2000.

[6] H. Duda, T. Gerlach, S. Advani and M. Potter. *Design of the DLR AVES Research Flight Simulator*. In: AIAA Modeling and Simulation Technologies (MS) Conference, Boston, MA, 2013.

[7] M. White and G. Padfield. *The Use of Flight Simulation for Research and Teaching in Acedemia*. In: AIAA Atmospheric Flight Mechanics Conference and Exhibit, Keystone, CO, 2006.

[8] S. Advani, D. Giovannetti and M. Blum. *Design of a Hexapod Motion Cueing System for NASA Ames Vertical Motion Simulator*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, California, 2002.

[9] O. Stroosma, R. van Paassen and M. Mulder. *Using the Simona Research Simulator for Human-Machine Interaction Research*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Austin, Texas, 2003.

[10] T. Longride, J. Bürki-Cohen, T. Go and A. Kendra. *Simulator Fidelity Considerations for Training and Evaluation of Today's Airline Pilots*. In: Proceedings of the 11th International Symposium on Aviation Psychology, Columbus, OH, 2001.

[11] D. Braun and R. Galloway, *Universal Automated Flight Simulator Fidelity Test System*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Rhode Island, 2004.

[12] C. Wang, J. He, G. Li and J. Han. *An Automated Test System for Flight Simulator Fidelity Evaluation*. Journal of Computers, vol. 4(11), 2009.

[13] C. Wang, J. Han, G. Li and H. Jiang. *Flight Simulator Fidelity Evaluation Automated Test System Analysis*. In: 2008 International Workshop on Education Technology and Training, Shanghai, China, 2008.

[14] P. Jarvis, D. Spira and B. Lalonde. *Flight Simulator Modeling and Validation Approaches and Pilot-in-the-loop Fidelity*. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Honolulu, Hawaii, 2008.

[15] J. Zander, I. Schieferdecker and P. Mosterman, A *Taxononomy of Model-Based Testing for Embedded Systems from Multipke Industry Domains.* In: Model-Based Testing for Embedded Systems, Boca Rato, CRC Press, 2012, pp. 3-23.

[16] A. Guduvan, H. Waselynck, V. Wiels, G. Durrieu, Y. Fusero and M. Schieber. *A Meta-Model for Tests of Avionics Embedded Systems*. In: Modelsward, Barcelona, Spain, 2013.

[17] D. Gasevic, D. Djuric and V. Devedzic. *Model Driven Architecture and Ontology Development*. Springer-Verlag, Berlin, 2006.

[18] T. Gruber. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Int. Journal of Human-Computer Studies, vol. 43, pp. 907-928, 1995.

[19] S. Weissleder. *Test Models and Coverage Criteria forAutomatic Model-Based Test Generation with UML State Machines*. Humboldt-Universität zu Berlin, Berlin, 2010.

[20] T. Moser, G. Düee and S. Biffl, *Ontology-Based Test Case Generation For Simulating Complex Production Automation Systems*. In: SEKE 2010, San Fransisco Bay, USA, 2010.

[21] C. Nguyen, A. Perini and P. Tonella. *Ontology-based Test Generation for Multiagent Systems.* In: 7th International Joint Conference on Autonomous Agents and Multiagent Systems, Estoril, Portugal, 2008.

[22] B. Zeigler. *Multifacetted Modelling and Discrete Event Simulation.* Academic Press Professional, Inc., 1984.

[23] B. Zeigler, H. Praehofer and T. Kim. *Theory of Modeling and Simulation: Integrating discrete event and continuous complex dynamic systems.* Academic Press, Inc., 2000.

[24] M. Traoré and A. Muzy. *Capturing the dual relationship between simulation models and their context.* Simulation Modelling Practice and Theory, 14, pp. 126-142, 2006.

[25] D. Foures, V. Albert and A. Nketsa. *Simulation Validation Using the Compatibility between Simulation Model and Experimental Frame.* In: 45th Summer Simulation Multi-conference, Toronto, Canada, 2013.

[26] K. Holger, M. Horridge, M. Musen, A. Rector, R. Stevans, N. Drummond, P. Lord, N. Noy, J. Seidenberg and H. Wangl. *The Protege OWL Experience.* In: OWLED, Galway, Ireland, 2005.

[27] P. Perfect, E. Timson, M. White, R. Erdos, A. Gubbels and A. Berryman. *A Rating Scale for Subjective Assesment of Simulator Fidelity*. In: 37th European Rotorcraft Forum, Gallarate, Italy, 2011.

[28] ICAO, *Manual Criteria for the Qualification of Flight Training Devices*. ICAO, Quebec, Canada, 2009.

[29] RAeS, *Aeroplane Flight Simulation Training Device Evaluation Handbook Vol.1 Objective Testing*. RAeS, London, 2009.

[30] B. Nader and J. B. Filippi, *An Experimental Frame for the Simulation of Forest Fire Spread*. In: Proceeding of the 2011 Winter Simulation Conference, Phoenix, Arizona, USA, 2011.

[31] A. Zengin and M. Ozturk, *Formal verification and validation with DEVS-Suite: OSPF Case study.* Simulation Modelling Practice, vol. 29, pp. 193-206, 2012.

[32] O. Corcho and A. Perez, *Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages*. In: ECAI'00 Workshop on Applications of Ontologies and Problem Solving Methods, Berlin, Germany, 2000.

[33] W. Borst, J. Akkermans, A. Pos and J. Top. *The PhysSys Ontology for Physical System.* In: QR'95 Ninth International Workshop on Qualitative Reasoning, Amsterdam, Netherlands, 1995.

[34] W. Borst and J. Akkermans. *Engineering Ontologies.* International Journal of Human-Computer Studies, vol. 46 (2/3), pp. 365-406, 1997.

[35] P. Fishwick and J. Miller. *Ontologies for Modeling and Simulation: Issues and Approaches*. In: Winter Simulation Conference, Washington, DC, 2004.

[36] U. Durak, H. Oguztuzun and K. Ider. *Ontology Based Trajectory Simulation Framework*. Journal of Computing and Information Science in Engineering, vol. 8(1), March 2008.

[37] U. Durak, H. Oguztuzun and K. Ider. *Ontology Based Domain Engineering for Trajectory Simulation Reuse*. Journal of Software Engineering and Knowledge Engineering, vol. 19(8), December 2009.

[38] B. Zeigler. *Modeling & Simulation-Based Data Engineering: Introducing pragmatics into ontologies for net-centric information exchange*. Academic Press, Inc., 2007.

[39] B. Zeigler and H. Sarjoughian. *Guide to Modeling and Simulation of Systems of Systems*. Springer, 2013.

[40] I. Harrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof and M. Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C, Canada, 2004.