TACKLING THE COMPLEXITY OF SIMULATION SCENARIO DEVELOPMENT IN AVIATION

Shafagh Jafer
Electrical, Computer, Software and Systems Eng.
Embry-Riddle Aeronautical University
600 S. Clyde Morris Blvd.
Daytona Beach, FL, USA
jafers@erau.edu

Umut Durak
Institute of Flight Systems
German Aerospace Center (DLR)
Lilienthalplatz 7
38108 Braunschweig, Germany
umut.durak@dlr.de

ABSTRACT

System complexity is a key characteristic in aviation industry which leads to broad utilization of modeling and simulation in in this global business. Scenario development is an important aspect of a simulation study. It starts at the very first steps when the operational scenarios are defined with the stakeholders and ends with a successful simulation execution. Although the importance of simulation scenarios has long been well-known, there still exists a lack of common understanding and standardized practices which lead to degraded interoperability and shareability. There is a recent effort coordinated by the American Institute of Aeronautics and Astronautics (AIAA) Modeling and Simulation Technical Committee (MSTC) towards development of a standard scenario definition language for aviation. This effort is being challenged by the same system complexity. Ontologies provide means to tackle complexity in domain modelling. This paper presents two distinct ontology based approaches to develop simulation scenario definition language for aviation. They both provide formal bases towards a standard domain specific language for scenario development.

Keywords: aviation, scenario definition, ontology, OWL, SES.

1 INTRODUCTION

A glance at the latest roadmaps from United States' Federal Aviation Administration (FAA) Next Generation implementation plan and aviation industry clearly reveal that future aircraft, airports, airlines, and air traffic management rely on advances in Cyber-Physical System (CPS) (Federal Aviation Administration 2016, Sampigethaya and Poovendran 2013). In aviation domain, CPS are extremely complex. A single flight of an aircraft from a departure to an arrival gate, for example, involves physical actuations of that aircraft's control surfaces, cyber domain interactions with ground, air, and space infrastructure, airborne sensing of natural processes such as weather, wind, and wildlife, and human-in-the-loop interactions such as between flight and ground crew. Simulation studies provide risk-free and low-cost environments to cope with the complexity of the aviation systems for defining high-level requirements, developing engineering designs, and expressing and enforcing security policies and training the operator.

Simulation scenarios are the subjects of interest throughout the whole simulation study. A simulation study typically starts with the description of the simulation scenario and ends with a successful simulation execution. Scenarios are not only used as executable specifications, but also utilized for identifying

SpringSim-MSCIAAS 2017, April 23-26, Virginia Beach, VA, USA

capability gaps, designing the new concepts of operations, minimum operational requirements and simulator features and even for testing the existing air and ground systems. Taking scenario development as one of the core simulation engineering activities, we require shared understanding, common practices among the stakeholders, operational subject matter experts and technical personnel and engineers, and we further need standards to boost interoperability and shareability. With this motivation, the American Institute of Aeronautics and Astronautics (AIAA) Modeling and Simulation Technical Committee (MSTC) recently launched a working group towards development of a standard scenario definition language for aviation.

Two of the important challenges when aiming at standardization in scenario development are to come up with a scenario definition language that will be natural to the scenario developers who are usually operational subject matter experts, and renders the required technical details to tackle the complexity of the aviation systems. Complexity comes from the large number of system/subsystems involved, deep hierarchical system structures, and complicated interrelations and inter-dependencies. Hence, scenario definition inevitably inherits this complexity. Furthermore, variability of intended use adds another flavor to complexity. It corresponds to variability in systems modeling both fidelity and resolution. As the structure and elements of the modeled air and ground systems vary, the simulation scenario definition also varies.

Ontologies have been proposed as the means for managing complex domain knowledge in the last two decades. The term ontology means a systematic explanation of existence in philosophy. Gruber defined it for computer science as: "Ontology is an explicit specification of a shared conceptualization" (Gruber 1995). We are investigating two ontology based approaches for developing a simulation scenario definition language. In the first one, we are building an ontology to capture the terms, concepts and their relations in aviation to support a simulation scenario definition language metamodel which is based on Base Object Model (BOM) (Durak et al. 2014; ASDL Ontology 2016; Jafer et al. 2016). In the second approach, we used System Entity Structures (SES) ontology in order to model the elements of a simulation scenario (Durak et al. 2017). This paper will first provide a background on simulation scenario development and ontologies. Then it will introduce these two efforts where ontologies are used to tackle the complexity of scenario development. It will finally conclude with a discussion about the comparison of these two approaches.

2 BACKGROUND

2.1 Simulation Scenario Development

One of the early definitions of simulation scenario is given in IEEE 1278, the Standard for Distributed Interactive Simulation, which introduces it as the description of initial conditions and timeline of significant events. The definition given in the High Level Architecture Glossary (US Department of Defense 1996) provides a more comprehensive description which states that a simulation scenario shall identify the major entities with their capabilities, behavior and interactions over time with all related environmental conditions. The elements of scenarios include the systems and subsystems of interest, their initial trim states, the environmental conditions, course of events through the simulation run and termination conditions.

Simulation scenarios are categorized into three types which are produced in successive stages of the scenario development process: operational scenarios, conceptual scenarios and executable scenarios (Siegfried et al. 2012; Siegfried et al. 2013; MSG-086 2014). Operational scenarios are usually described in the early stages of a simulation study in a textual form. They typically provide a coarse description of the intended situation and its dynamics, they require refinement and enhancement by the simulator experts with additional information pertaining to simulation. That effort leads to conceptual scenarios that are complete and consistent. The executable scenario is then the specification of the conceptual scenario in a particular format in order to be processed by the simulation applications.

AIAA MSTC workgroup on simulation scenario development aims at standardization of conceptual scenario development. A well-formed standard language for specifying conceptual simulation scenarios in aviation is promoted as the key for the following goals:

- sharing scenario definitions across various facilities and experts,
- boosting interoperability of scenario definition tools and execution infrastructures,
- increasing the quality of scenarios by formal completeness and consistency checking approaches
- enable automated scenario generation using language constructs

2.2 Simulation Scenario Definition Language

Developing a scenario definition language for a specific domain has already been conducted for military simulations. Military Scenario Definition Language (MSDL) was developed and published as a standard by Simulation Interoperability Standards Organization (SISO) (SISO 2008).

The simulation scenario definition language is being proposed as a Domain Specific Language (DSL) for scenario specification in aviation domain. DSLs are custom tailored computer languages for a particular application domain that specifically target problems in that specific domain, and stresses upon the main ideas, features, constraints, and characteristics of that domain (Brambilla, Cabot und Wimmer 2012).

Besides a standardized language, it is crucial to develop methods and techniques as well as reference implementations of tools like editor and compiler to raise the level from an academic one to a well usable language in the field. This will support the acceptance within the simulation community and lower the threshold for users as well as establishing an ecosystem around the scenario definition language.

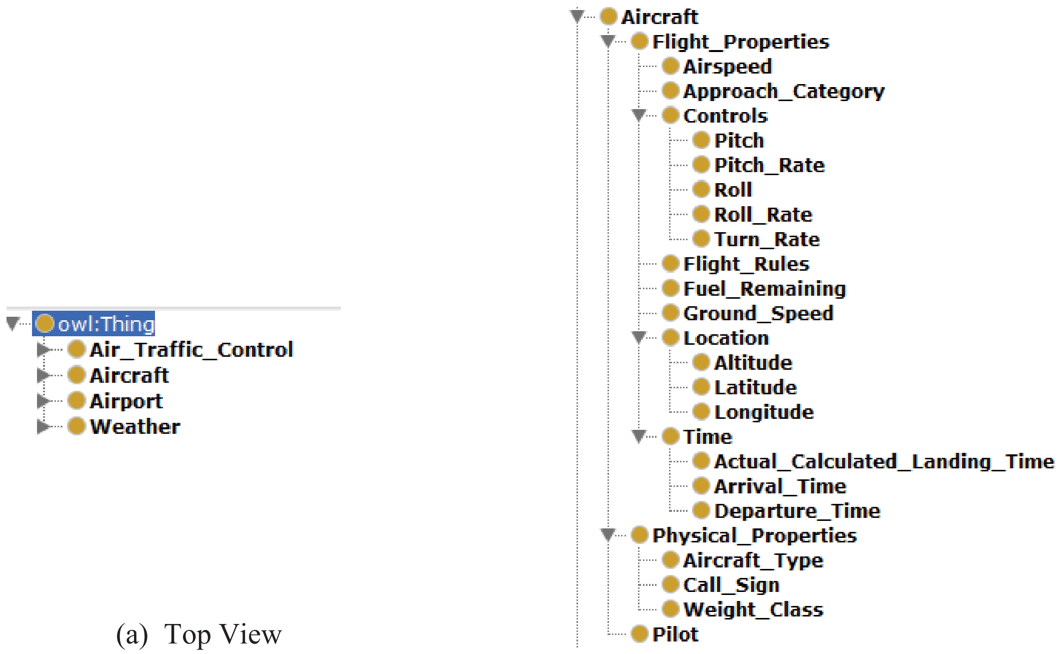
2.3 Ontologies in Domain Modeling

Domain modeling aims at creating a unified resource as a repository of shared knowledge in the domain of interest (Arango und Prieto-Diaz 1991). This unified resource is the baseline of a DSL. The challenge of domain modeling is representing this knowledge which essentially inherits all the complexity of the domain for ease of human understanding and machine processing (Prieto-Diaz 1990). Ontologies help us at this point describe the key concepts and relationships in a particular domain, providing a vocabulary for that domain as well as a computerized specification of the meaning of terms used in the vocabulary in both machine and human readable form (Yao and Zhang 2009).

Ontology-driven domain modeling has emerged as a significant mechanism in developing domain-specific languages (Čeh et al. 2011; Pan, et al. 2012), providing an effective approach for expressing domain concepts. Ontology provides a quick and simplified description of a DSL, abstracting language’s technically details, while highlighting key terminology and specifics. Among the existing ontology specification frameworks, the Web Ontology Language format (OWL) is most commonly used by the DSL community. OWL enables describing a domain in terms of classes, properties and individuals and may include rich descriptions of the characteristics of those objects (Bechhofer 2009).

3 AVIATION SCENARIO DEFINITION LANGUAGE (ASDL) ONTOLOGY

In order to capture aviation scenario details, it is essential to have a definitions reference list that highlights all key terminology as well as procedures and operations that are communicated during a flight mission. The FAA’s NextGen and Single European Sky ATM Research SESAR programs provide inclusive glossaries that provide key terminology and concept of operations (Federal Aviation Administration 2012; SESAR 2015). A complete ASDL ontology was developed making use of these terminologies in OWL using Protégé (Alatrish 2013; Protégé Home Page 2016). An OWL ontology describes a domain in terms of classes, properties and individuals and may include rich descriptions of the characteristics of those objects.



(a) Top View

(b) Aircraft

Figure 1: ASDL Ontology.

An ontology focuses mainly on classes which describe the concepts of the domain. It follows a hierarchical model where subclasses are all necessarily a part of the superclass (Noy and McGuinness 2001). Accordingly, ASDL ontology has four base classes: Air_Traffic_Control, Aircraft, Airport and Weather as depicted in Figure 1 (a). All these terms have been defined in Table 1. Figure 1 (b) presents an excerpt, namely the Aircraft class. Its subclasses are Flight_Properties, Physical_Properties, and Pilot. Flight_Properties further decomposed into its subclass such as the rules (Instrument Flight Rule (IFR) or Visual Flight Rule (VFR)) that govern aspects of the flight such as the speed of the aircraft. Details on current ASDL ontology can be found in (ASDL Ontology 2016)

Table 1: Definition of terms in base class of ASDL Ontology.

Term	Definition
Air Traffic Control	A service operated by appropriate authority to promote the safe, orderly and expeditious flow of air traffic.
Aircraft	Any machine that can derive support in the atmosphere from the reactions of the air other than the reactions of the air against the earth's surface.
Airport	An area on land or water that is used or intended to be used for the landing and takeoff of aircraft and includes its buildings and facilities, if any.
Weather	The state of the atmosphere at a place and time as regards heat, dryness, sunshine, wind, rain, etc.

ASDL adopts Model-Driven Engineering (MDE) to develop scenarios following the principles of MDE, scenario development takes place as the transformation of operational scenarios (defined in a natural language) to conceptual scenarios (conforming to ASDL formal metamodel) via modeling then to executable scenarios (specified using ASDL scenario definition) using model transformations. To capture all the necessary constructs for a simulation scenario, SISO BOM was adopted as the baseline metamodel adapting the ideas in (Durak et al. 2014). BOM is a standard that introduces the interplay, the sequence of events between simulation elements, as well as the reusable pattern, and provides a standard to capture the interactions. In ASDL, this baseline was extended to capture all the domain related concepts and terminology as constructs. The Scenario class is the base class that contains all the other classes in the metamodel. An instance created of this class is used to define all other objects and their attributes. The current ASDL model object allows users to define three different kinds of scenarios: departure, rerouting and landing. It also includes pilots, airports, runways, control towers, flight properties, weather patterns and aircrafts. This metamodel was integrated with the BOM entities of interplays, state machines and events in order to describe a flight scenario.

4 MODELING SCENARIO ELEMENTS WITH SYSTEM ENTITY STRUCTURE

System Entity Structure (SES) is a high-level ontology which has its roots in system theory-based approach to modeling and simulation. It was introduced for knowledge representation of decomposition, taxonomy and coupling of systems (Kim et al. 1990).

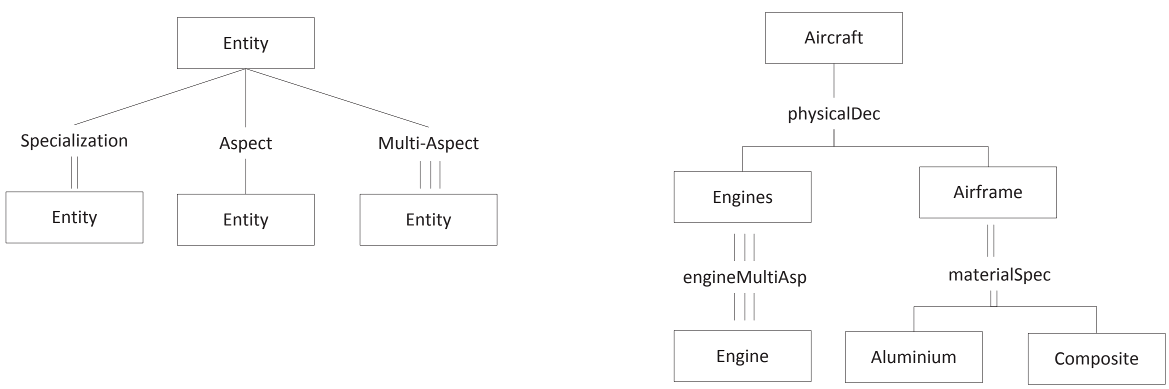


Figure 2: SES Nodes and relationships and an Example.

SES can also be introduced as a directed and labelled tree composed of Entity and Aspect, Specialization and Multi-Aspect nodes (Figure 2). Entity is an object of interest. Variables can be attached to Entities. Aspect denotes the decomposition relationship of an entity while specialization represents its taxonomy. Aspects are represented by vertical lines and Specializations with double line. The Multiple-Aspect is a special kind of Aspect that represents a multiplicity relationship that specifies that the parent entity is a composition of multiple entities of the same type. Three vertical lines are used to represent Multiple Aspects.

In addition to these four basic elements, it possesses six clearly defined axioms which are named as uniformity, strict hierarchy, alternating mode, valid brothers, attached variables and inheritance (Zeigler 1984). Uniformity states that any two nodes with the same labels have isomorphic subtrees. Strict hierarchy prevents a label from appearing more than once down any path of the tree. Alternating mode endorses that, if a node is an Entity, then the successor is either Aspect or Specialization, and vice versa. Valid siblings prohibits having two siblings with the same label. Attached variables state that variable types attached to the same item shall have distinct names. With inheritance, it is stated that Specialization inherits all variables and Aspects.

Pruning is defined as resolving the choices in Aspect, Multi-Aspect and Specialization relations and assigning the values to the variables. SES may have several Aspect nodes for several decomposition on the same hierarchical level, a particular one needs to be chosen in pruning based on the purpose. In Specializations nodes one Entity out of various variants needs to be selected. The cardinality in Multi-

Aspect relations is also specified in pruning, and a composition needs to be constructed with valid brother of the same type of entities with distinct identifications. The result is call Pruned Entity Structure (PES), which is a selection-free tree.

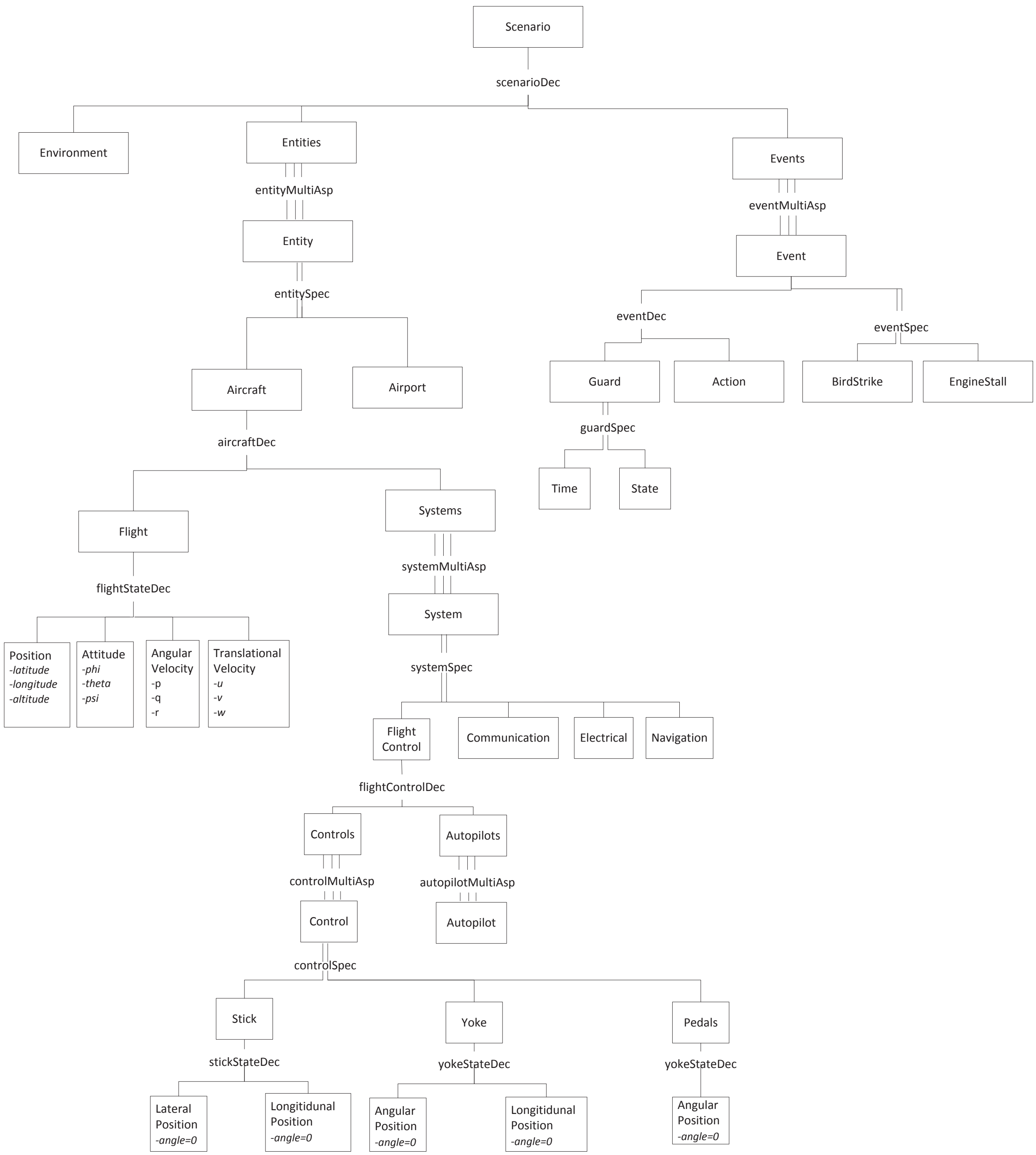


Figure 3: Scenario Metamodel Extract.

SES is regarded as a promising ontological foundation for tackling the complexity in simulation scenario development regarding its clear and simple structure and it capability to model deep hierarchical system and data structures with variability (Zeigler and Hammonds 2007).

With a model-driven approach, scenario definition language enables the definition of a concrete representation for a scenario model where we require a metamodel to define the scenario definition language (Brambilla, Cabot und Wimmer 2012). We utilize SES for metamodeling in order to capture all possible meta elements of simulation scenarios in aviation. While the Aspect and Multi-Aspect nodes enables us to capture compositional hierarchy of scenario and Specialization nodes are used to manage variability. Further, based on the extensions to SES which are proposed by Pawletta et.al (Pawletta et al. 2014), SES functions, selection rules/constraints on Aspect, Multi-Aspects and Specialization nodes are regarded as means to capture the dependencies between scenario elements and semantic constraints.

Scenario definition is then using simulation scenario SES, and pruning it to specify a particular scenario. Pruning can be accomplished either manually in an interactive manner or via automated means using a scripting front-end. SES functions are evaluated during pruning and selection rules/constraints are employed for enforcing dependencies or checking constraints.

A representative excerpt of the AVES Scenario Metamodel is presented in Figure 3. Scenario entity is decomposed at the top level using the scenarioDec aspect node into Environment, Entities and Events which are the main parts of a simulation scenario. entityMultiAsp multi-aspect node denotes that there can be more than one entities Entities. Different types of Entity are then captured using entitySpec specialization node. Two representative Entity nodes from a larger set are Aircraft and the Airport. Aircraft is decomposed using aircraftDec in to its aspects, Flight and Systems, that are of interest as elements of a scenario. Flight is then decomposed into its states: Position, Attitude, Angular Velocity and Translational Velocity. The common system decomposition of an aircraft into its subsystems is done using systemSpec based on FAA Joint Aircraft System/Component Code Table and Definition (Federal Aviation Administration 2008). The selected subset that is depicted in the figure includes Flight Control, Communication, Electrical and Navigation. FlightControl is composed of Controls and Autopilots, further down Stick, Yoke and Pedals are types of Control. As the leaf nodes, Lateral Position and Longitudinal Position are the data elements of a scenario concerning a Stick. Attached variables of the leaf Entity nodes correspond to a scenario data to capture the data values. Lateral Position entity has a variable called angle. Some of the variables may have default values which can also specified be at the SES, such as Lateral or Longitudinal Positions of the Stick which are by default set to zero.

Event is composed of a Guard and an Action. Guards are logical statements. Two Guard types are State and Time, which designates either a data item that describes the simulation state or the simulation time is used in specifying them. eventSpec is on the other side is used to capture various types of Event. Examples are Bird Strike and Engine Stall.

An example use case of SES functions and selection rules/constraints can be given with the Bird Strike event. An SES function can be used to identify selection of such an event and setting a particular selection rule/constraint so that it guarantees the incorporation of flock of birds to the scenario.

5 DISCUSSION AND CONCLUSION

Ontologies provides us invaluable means to tackle complexity in knowledge representation. Accordingly, ontology-driven software development has gained significant interest in domain-specific application development. To tackle the aviation scenario definition problem, building an ontology was the first step towards the design of an Aviation Domain Specific Language, as it helps define the classes, attributes and relationships which describe the domain. In this research, we investigated two different ontology based approaches. In the first one we utilized widely used Web Ontology Language (OWL) and Protégé tool. In the second one, we promoted System Entity Structures (SES) as the ontological foundations of domain modeling.

Both approaches have their strengths and weaknesses. On one hand, OWL provides a more expressive formal basis for ontology construction. Additionally Protégé is a mature tool with a wide community support. But both Protégé and OWL are not natural to scenario developers. Formal semantics of OWL is based on description logic which is out of their body of knowledge. On the other hand, SES with its simple formal semantics which is based on systems theory can be easily grasped and used by scenario developers. However, the there is a lack of mature tool sets and common techniques for SES development. Its application out of modeling and simulation academia is very limited. Therefore it is hard to talk about a community support.

The working group on simulation scenario development coordinated by the American Institute of Aeronautics and Astronautics (AIAA) Modeling and Simulation Technical Committee (MSTC) will be the platform to discuss approaches similar to what we presented in this paper with representatives from the industry, research labs, academia and regulatory authorities. The objective is to come up with a standard scenario definition language in aviation domain in the near future.

REFERENCES

Alatrish, E. 2013. "Comparison Some of Ontology Editors." *Management Information Systems* vol. 8, no. 2 , pp. 018-024.

Arango, G., and R. Prieto-Diaz. 1991. "Domain Analysis Concepts and Research Directions." In *Domain Analysis and Software Systems Modeling*, by G. Arango and R. Prieto-Diaz, 9-31. IEEE Computer Society Press.

ASDL Ontology 2016. "Aviation Scenario definition Language Ontology". https://github.com/ASDL-prj/Ontology Accessed Feb. 07 2017.

Bechhofer, S. 2009. "OWL: Web Ontology Language." In *Encyclopedia of Database Systems*, by Ling Liu and M. Tamer Özsu, Boston, MA: Springer US.

Brambilla, M., J. Cabot, and M. Wimmer. 2012. *Model-Driven Software Engineering in Practive.* London, UK: Morgen & Claypool.

Čeh, I., M. Črepinšek, T. Kosar, and M. Mernik. 2011. "Ontology Driven Development of Domain-Specific Languages." *Computer Science and Information Systems* vol. 2, pp. 317-342.

Durak, U., I. Pruter, T. Gerlach, S. Jafer, T. Pawletta, and S. Hartmann. 2017. "Using System Entity Structures to Model the Elements of Scenario in a Research Flight Simulator." *AIAA Modeling and Simulation Technologies Conference.* Grapevine, TX.

Durak, U., O. Topcu, R. Siegfried, and H. Oguztuzun. 2014. "Scenario Development: A Model-Driven Engineering Perspective." *2014 International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH).* Vienna, Austria.

Federal Aviation Administration 2008. *Federal Aviation Administration Joint Aircraft System/Component Code Table and Definitions.* Oklahoma City, OK.

Federal Aviation Administration. 2016. NextGen Implementation Plan. Washington, DC, Available: https://www.faa.gov/nextgen/media/NextGen_Implementation_Plan-2016.pdf Accessed Feb. 7, 2017.

Federal Aviation Administration 2012. *AFS Flight Program Flight Operations Manual.* Oklahoma City, OK.

Gruber, T. R. 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing" *International Journal of Human-Computer Studies* vol.43, no. 5, pp.907-928.

Jafer, S., B. Chhaya, U. Durak, and T. Gerlach. 2016. "Formal Scenario Definition Language for Aviation: Aircraft Landing Case Study." *AIAA Modeling and Simulation Technologies Conference.* Washington, DC.

Kim, T.G., C. Lee, E.R. Christensen, and B.P. Zeigler. 1990. "System Entity Structuring and Model Base Management." *IEEE Transactions on Systems, Man, and Cybernetics*, vol.20, no.5 pp. 1013-1024.

MSG-086 2014. "Guideline on Scenario Development for (Distributed) Simulation Environments". STO Technical Report, Brussels, Belgium.

Noy, N.F., and D.L. McGuinness. 2011. "Ontology Development 101: A Guide to Creating your first Ontology." Available via http://protege.stanford.edu/publications/ontology_development/ontology101.pdf. Accessed Feb. 7, 2017.

Pan, J.Z., S. Staab, U. Aßmann, J. Ebert, and Y. Zhao. 2012. *Ontology-driven software development.* Springer Science & Business Media.

Pawletta, T., D. Parscheka, A. Schmidt, and S. Pawletta. 2014. "Ontology-Assisted System Modeling and Simulation within MATLAB/Simulink." *Simulation Notes Europe* vol.24, no. 2, pp. 59-68.

Prieto-Diaz, R. 1990. "Domain Analysis: An Introduction." *ACM SIGSOFT Software Engineering Notes* col.15, no.2, pp. 47-54.

Protégé Home Page. 2016. "Protégé Hope Page" http://protege.stanford.edu/. Accessed Feb. 7, 2017.

Sampigethaya, K., and R. Poovendran. 2013. "Aviation Cyber-Physical Systems: Foundations for Future Aircraft and Air Transport." *Proceedings of the IEEE* vol.101, no.8, pp. 1834-1855.

SESAR 2015. *EUROCONTROL ATM Lexicon.* Single European Sky ATM Research.

Siegfried, R. A. Laux, M. Rother, D. Steinkamp, G. Herrmann, J. Lüthi, and M. Hahn. 2012. "Scenarios in Military (distributed) Simulation Environments." *Spring Simulation Interoperability Workshop (S-SIW).* Orlando, FL.

Siegfried, R., H. Oguztüzün, U. Durak, A. Hatip, G. Herrmann, P. Gustavson, and M. Hahn. 2013. "Specification and Documentation of Conceptual Scenarios Using Base Object Models (BOMs)." *Spring Simulation Interoperability Conference.* San Diago, CA.

SISO 2008. "Standard for Military Scenario Definition Language (MSDL)". Standard, Orlando, FL.

US Department of Defense 1996. "High Level Architecture Glossary." Washington, DC.

Yao, Z., and Q. Zhang. 2009. "Protégé-Based Ontology Knowledge Representation for MIS Courses ." *International Conference on Web Information Systems and Mining.* Shangai, China.

Zeigler, B.P. 1984. *Multifaceted Modeling and Discrete Event Simulation.* London, UK, Academic Press Professional Inc..

Zeigler, B.P., and P.E. Hammonds. 2007. *Modeling and Simulation-Based Data Engineering: Introducting Pragmatic into Ontologies for Net-Centric Information Exchange.* London, UK, Academic Press.

AUTHOR BIOGRAPHIES

SHAFAGH JAFER is an Assistant Professor of Software Engineering at Embry-Riddle Aeronautical University. She holds a PhD in Computer Systems Engineering from Carleton University in Canada. Her research interest include simulation-based engineering, mission-critical software systems, and model-based development. Her email address is jafers@erau.edu.

UMUT DURAK is a Research Scientist at Institute of Flight Systems of German Aerospace Center (DLR). He holds a PhD in Mechanical Engineering from Middle East Technical University in Turkey. His research interests include model-driven approaches applied to engineering of simulation systems and modeling and simulation based engineering of flight systems. His email address is umut.durak@dlr.de.