Human Systems Engineering and Design (IHSED 2024), Vol. 158, 2024, 211–219
https://doi.org/10.54941/ahfe1005544

Importance of Ontologies for Systems Engineering (SE) and Human Factors Engineering (HFE) Integration

Sarah Rudder[1,2] **and Daniel R. Herber**[1]

[1]Colorado State University, Fort Collins, CO, USA
[2]Enola Technologies, USA

ABSTRACT

The integration of Systems Engineering (SE) and Human Factors Engineering (HFE) is essential for developing comprehensive and reliable systems. This paper addresses the challenge of disparate vocabularies between these domains by leveraging ontologies to facilitate a common understanding. Utilizing standard practices and tools, the creation and implementation of ontologies are demonstrated through this research. By establishing a shared language and linking domain-specific terms, this work highlights the potential of ontological virtual models to enhance stakeholder communications.

Keywords: Systems engineering, Human factors engineering, Ontology, Knowledge sharing, Data integration

INTRODUCTION

This research is motivated by the need to overcome barriers created by disparate vocabularies between domains, approaches, representations, and tools (Uschold & Jasper, 1999). Shani et al., 2016, state that leveraging the knowledge of relationships between information reduces costs and risks, improves designs, and shortens schedules. This work supports the International Council on Systems Engineering (INCOSE) Systems Engineering Vision 2035 goal of data integration between multiple tools.

This paper aims to answer the following research questions: (1) How can SE and HFE ontologies be defined? (2) How do these ontologies enable the mapping of fundamental terms across domains?

Ontologies

Systems engineering (SE) must implement human factors engineering (HFE) to properly define user interfaces and satisfy human-system integration (HSI) requirements. For individuals with diverse backgrounds and experience levels, a documented ontology proves useful for a baseline understanding of synonymous terms that cross disciplines.

An ontology is an explicit specification of a common language described by a set of representational terms (Gruber, 1993). These are logical theories encoded using knowledge representation languages (Aminu et al., 2020). Ontological models define natural language in a machine-readable format (Mkhinini et al., 2020). To accurately exchange information, each relevant entity and relationship must be identified and explicitly defined (van Ruijven, 2015). Every entity defined by a formal ontology is studied by multiple disciplines, (Munn & Smith, 2013), which contributes to the challenges of defining object-specific transformation as an extension of the model (Lütjen et al., 2014). The interdisciplinary nature of adopting a modeling approach creates language barriers between stakeholders (Lütjen et al., 2014) that are overcome with a unanimous understanding of shared terminology. The formalization of ontologies emphasizes the machine-readable structured repository (Aminu et al., 2020). Additionally, ontologies include concept definitions that indicate domain structure and constrain possible interpretations of the language (Gruber, 1993).

Ontologies are viewed as the interface between the knowledge base and reality that guides information shareability, acquisition, and organization (Kang et al., 2010). According to Noy & McGuinness, 2001, reasons for ontology development include:

- To share common understanding
- To enable reuse of domain knowledge
- To explicitly define domain assumptions
- To analyze domain knowledge.

Basic Foundational Ontology (BFO)

The basic foundational ontology (BFO) is a high-level ontology developed and designed to represent common categories of domain-specific languages (Arp & Smith, 2008). This standard comprises of (1) continuants: entities that continue or persist through time, and (2) occurrents: the events in which continuants participate (Arp & Smith, 2011).

Web Ontology Language (OWL)

According to the web ontology language (OWL) v2 specification developed by the World Wide Web Consortium (W3C), ontologies are formalized vocabulary terms for a specific domain that are shared among users. OWL ontologies map to resource description framework (RDF) graphs and include annotations of classes and properties. OWL can enhance both precision and accessibility, but it requires mediation to enhance model semantics and resolve conflicts (Kulvatunyou, et al., 2014). Mediator systems federate and integrate data from disparate sources to elicit information that cannot be provided by an individual source (Ludascher, et al., 2001). The Semantic Mediation Container (SMC) assumes that all RDF models are associated with an ontology that provides the semantics of the graph elements (Shani, 2016).

Resource Description Framework (RDF)

The RDF is used by all OWL syntaxes to provide a common approach for expressing information to prevent data exchange between software applications from losing meaning (W3C, 2004). Each entity is given a unique

Internationalized Resource Identifier (IRI) to ensure statements are machine-readable. Using graph theory, subjects and objects are represented as nodes (i.e., vertices) while predicates are shown as paths (i.e., edges) between the nodes (Fournier, 2009). RDF defines all data as triples composed of a subject, predicate, and object [SPO] (Ernadote, 2015). Therefore, the SPO approach aligns with a graph-based data model profile that uses nodes and edges to convey domain structure as shown in Figure 1.

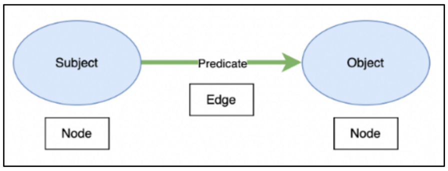


Figure 1: SPO data model representation.

RESEARCH METHOD

The SPO methodology for building domain-specific ontologies links subjects and objects together with predicates. A modified version of SPO was introduced for these ontological models to incorporate the inverse predicates for each triple shown with a bi-directional relationship in Figure 2. In addition to the adapted relationship, subjects and objects are both represented by *classes*, which according to Noy & McGuinness (2001), represent concepts in the domain.

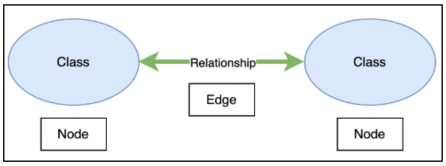
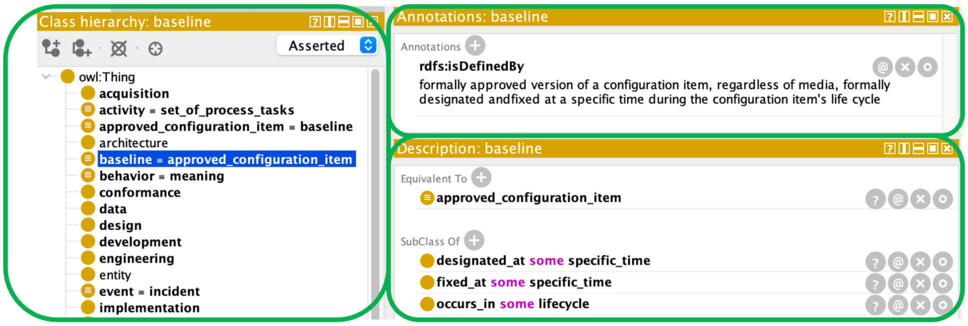


Figure 2: SPO data model representation with bi-directional predicates.

The open-source Protégé tool v5.6.4 was used as an ontology editor since it supports both continuant and occurrent entity semantics. Protégé defines classes and class hierarchies, variables and value restrictions, and the relationships between classes with properties (Malviya et al., 2011). Although several ontology editing tools support knowledge engineering integrated with semantic reasoners, the Protégé user interface proved most conducive to this research.

To create the International Organization of Standards (ISO) 15288 - *Systems Engineering Processes* ontology, a middle-out approach was used for development, where the most important concepts are captured and followed by related concepts (Aminu et al., 2020). The twenty (20) most used nouns in the standard provided the foundation for the initial ontology iteration.

The National Aeronautics and Space Administration (NASA) HSI handbook was chosen to design the first iteration of the HSI ontology since the literature satisfies the requirements of this research by considering both SE and HFE concepts. A top-down approach first identified the most abstract concepts and then specified more detailed concepts (Aminu et al., 2020). The top twenty (20) mentioned nouns were evaluated and filtered to represent high-level, abstract terms. Vocabulary without explicit definition in the HSI handbook glossary and absent from Section 3 of ISO 15288 were assumed to be too concrete for inclusion in the initial iteration of this ontology. Figure 3 shows the Protégé user interface and the fields that capture classes along with associated annotations and relationships.



owl:Thing
acquisition
activity ≡ set_of_process_tasks
approved_configuration_item ≡ baseline
architecture
baseline ≡ approved_configuration_item
behavior ≡ meaning
conformance
data
design
development
engineering
entity
event ≡ incident
implementation

rdfs:isDefinedBy
formally approved version of a configuration item, regardless of media, formally designated andfixed at a specific time during the configuration item's life cycle

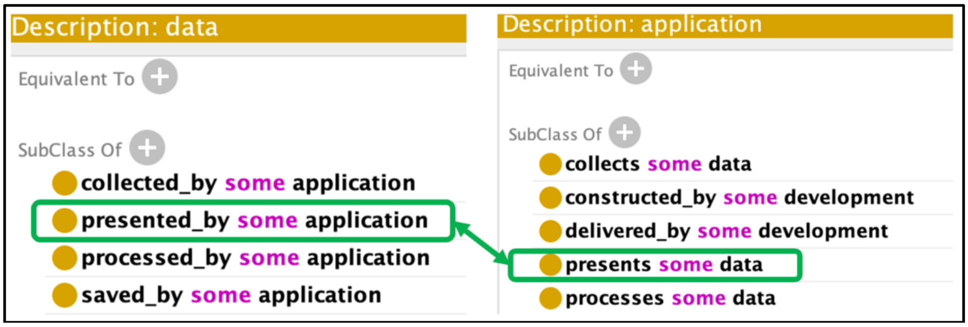
Equivalent To
approved_configuration_item

SubClass Of
designated_at some specific_time
fixed_at some specific_time
occurs_in some lifecycle

Figure 3: Example of class hierarchy, annotations, and description in Protégé.

After this initial task, the bi-directional associations were added to the *object property hierarchy* in Protégé. These verb phrases are assigned as predicates between two classes to define relationships between subjects and objects.

Figure 4 illustrates the predicates assigned to the *data* class (left) of *collected by*, *presented by*, etc. The inverse predicate of the *data* class relations are assigned to the *application* class (right) as *collects*, *presents,* etc. These triples are read as “data is presented by some application”, where *data* is the subject, *presented by* is the predicate, and *application* is the object.



collected_by some application
presented_by some application
processed_by some application
saved_by some application

collects some data
constructed_by some development
delivered_by some development
presents some data
processes some data

Figure 4: Predicates (left) and inverse predicates (right) in Protégé.

RDF graphs were created using the Ontograf plugin for Protégé to enable a richer understanding of SPO triples and the interconnectedness of terms and relationships. The visualizations represent *classes* as nodes and *predicates* as arcs shown in Figure 5.

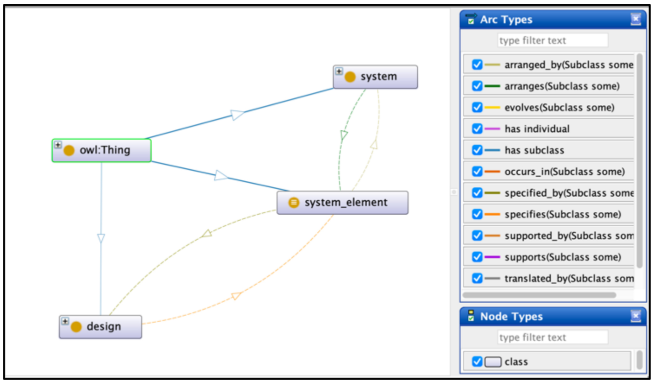


Figure 5: Example of RDF visualization for SPO triples.

Transferring OWL files between Protégé and Cognitum FluentEditor was conducted to validate accurate data translation of SPO triples. *Classes* defined in Protégé were successfully transformed into *things* and *predicates* to *relations* within FluentEditor, ensuring the output of Protégé is selectable. Results of the class hierarchy import are shown in Figure 6.

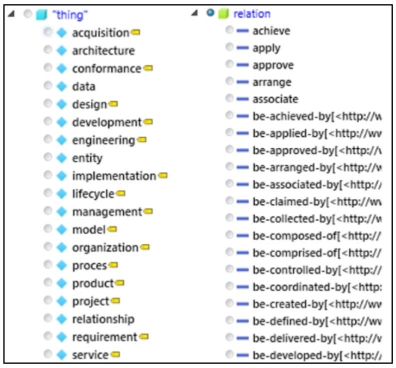
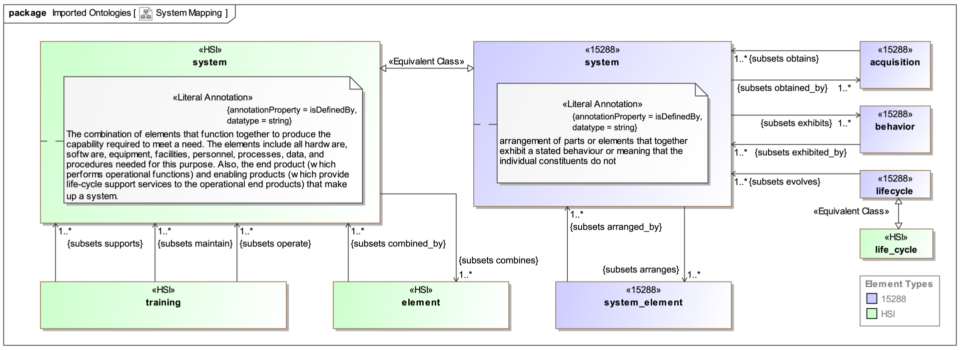


Figure 6: Cognitum FluentEditor data transformation from Protégé.

OWL files were then imported into a model-based systems engineering (MBSE) tool for conceptual modeling. Conceptual models are high-level abstractions that enable stakeholders to communicate with each other (Embley & Thalheim, 2011). While ontologies define terms that represent knowledge, a conceptual schema defines relations between data types [i.e., subjects, objects, "things"] (Gruber, 1993). Mapping synonyms between domains is facilitated by conceptual modeling for understanding high-level abstractions of terms, enabling stakeholders to unambiguously communicate with each other (Embley & Thalheim, 2011).

Both ontologies were imported into a single BFO concept model to map the definitions and relations. The term *system* was in the twenty (20) most used terms in the NASA HSI handbook. An *equivalent class* relationship in the concept model between the *system* class from the ISO 15288 ontology and the *system* class from the HSI ontology is shown in Figure 7.

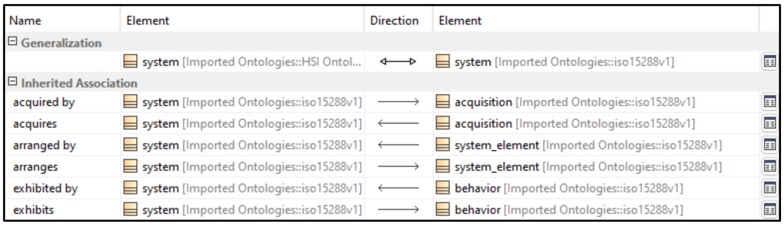


The combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, data, and procedures needed for this purpose. Also, the end product (which performs operational functions) and enabling products (which provide life-cycle support services to the operational end products) that make up a system.

arrangement of parts or elements that together exhibit a stated behaviour or meaning that the individual constituents do not

Figure 7: Concept model diagram mapping HSI ‘system’ class to ISO 15288 vocabulary and relationships.

The SPO triples between the ISO 15288 *system* class, behavior, and its elements were automatically populated based on the OWL file import into the MBSE tool. The equivalent terms identified in Protégé for *behavior* and *system element* (i.e., *part* and *meaning*) were imported from the OWL file. Based on the concept of inheritance between generalized terms, all relationships applied to the ISO 15288 *system* are *inherited associations* of the HSI *system* class shown in Figure 8. This example demonstrates how ontologies map to each other, filling knowledge gaps of ambiguous terms. The ISO 15288 *system* class simultaneously inherits all associations from the HSI *system* class (specification not shown).



Name	Element	Direction	Element
⊟ Generalization			
	system [Imported Ontologies::HSI Ontol...	↔	system [Imported Ontologies::iso15288v1]
⊟ Inherited Association			
acquired by	system [Imported Ontologies::iso15288v1]	→	acquisition [Imported Ontologies::iso15288v1]
acquires	system [Imported Ontologies::iso15288v1]	←	acquisition [Imported Ontologies::iso15288v1]
arranged by	system [Imported Ontologies::iso15288v1]	←	system_element [Imported Ontologies::iso15288v1]
arranges	system [Imported Ontologies::iso15288v1]	→	system_element [Imported Ontologies::iso15288v1]
exhibited by	system [Imported Ontologies::iso15288v1]	←	behavior [Imported Ontologies::iso15288v1]
exhibits	system [Imported Ontologies::iso15288v1]	→	behavior [Imported Ontologies::iso15288v1]

Figure 8: Properties inherited by the HSI ‘system’ class from the ISO 15288 ‘system’ class.

ISO 15288 and the NASA HSI handbook give the term *baseline* different definitions. The *equivalent class* relationship forms a more holistic understanding of the term and its true meaning shown in Figure 9.

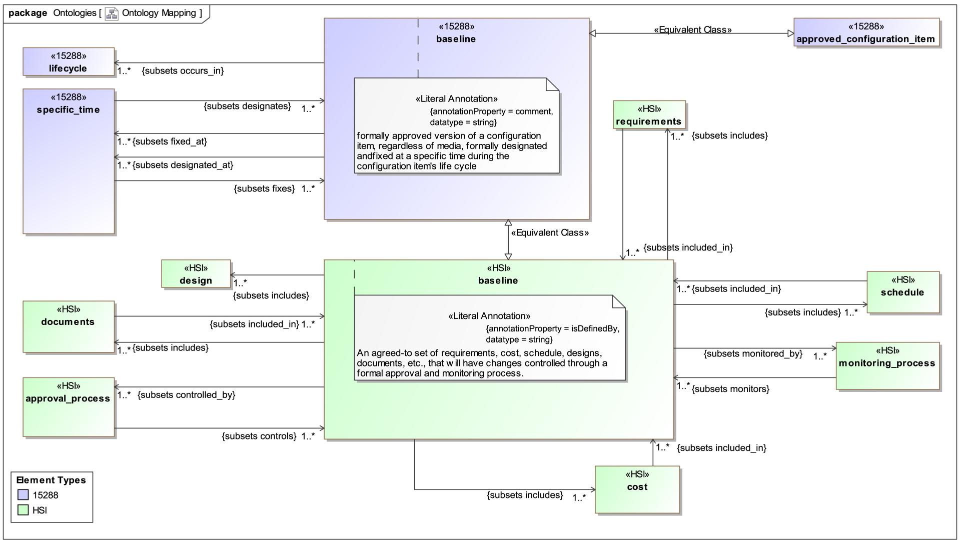


Figure 9: ISO 15288 and NASA HSI definition of ‘baseline’.

The following describes the scenario where similar concepts have different terms assigned to them. For example, ISO 15288 defines an *incident* as an “*anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system*”. The NASA HSI handbook defines a *critical event* as “*an event in the operations phase of the mission that is time-sensitive and is required to be accomplished successfully in order to achieve mission success*”. The likeness in these descriptions supports establishing an *equivalent class* relationship between the terms. Figure 10 demonstrates mapping vocabulary within the conceptual model for this use case.

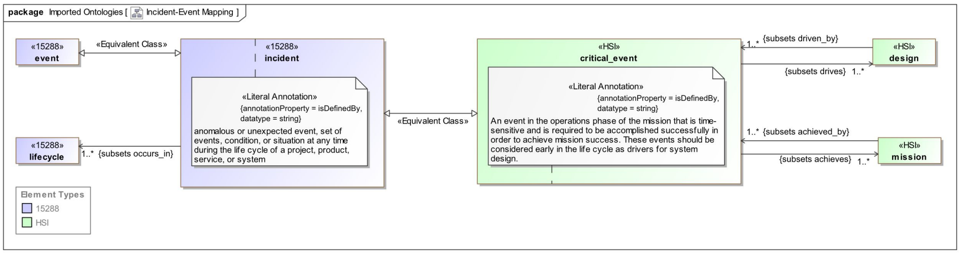


Figure 10: Relationship between ISO 15288 and NASA HSI like-terms.

RESEARCH CONCLUSIONS

This paper gives an example of initial domain-specific ontology iterations, SPO triples, and equivalent class mapping within a digital environment. The fundamental terms for both SE and HFE definitions captured in Protégé represented the initial conclusion for research question (1). The conclusion for research question (2) is supported by the import of multiple ontologies into a single digital concept modeling environment for *equivalent classes* to relate vocabulary terms between several domains.

This research contributes to the goals of the INCOSE Systems Engineering Vision 2035; including routine composition of domain-specific virtual models using ontologically linked assets. These new “digital dictionaries” provide an authoritative source of truth that defines equivalent classes between domain-specific terms. However, this research has several limitations. Currently, there is a lack of iterations to resolve semantic conflicts. Second, the presented initial ontologies are not mature enough for industry application.

FUTURE RESEARCH

Future research includes incremental and recursive iterations of the vocabulary in both ISO 15288 and the NASA HSI handbook for more complete ontologies. Once these repositories are built-out, structured expressions for querying the data will be executed to determine the usefulness of the linked languages. By focusing on the refinement of these ontologies, the data integration process will be further improved and implemented for a case study to explore the potential reduction of costs and risks with this approach.

REFERENCES

Arp, R., & Smith, B. (2008). Function, Role, and Disposition in Basic Formal Ontology. *Nature Precedings*, 1–1. https://doi.org/10.1038/npre.2008.1941.1

Arp, R., & Smith, B. (2011). *Realizable Entities in Basic Formal Ontology*.

Dori, D. (2024). Model-based standards authoring: ISO 15288 as a case in point. *Systems Engineering*, *27*(2), 302–314. https://doi.org/10.1002/sys.21721

Embley, D. W., & Thalheim, B. (Eds.). (2011). *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-15865-0

Ernadote, D. (2015). *An ontology mindset for system engineering* (p. 460). https://doi.org/10.1109/SysEng.2015.7302797

Femi Aminu, E., Oyefolahan, I. O., Bashir Abdullahi, M., & Salaudeen, M. T. (2020). A Review on Ontology Development Methodologies for Developing Ontological Knowledge Representation Systems for various Domains. *International Journal of Information Engineering and Electronic Business*, *12*(2), 28–39. https://doi.org/10.5815/ijieeb.2020.02.05

Fournier, J. C. (2009). *Graph Theory and Applications: With Exercises and Problems*. https://doi.org/10.1002/9780470611548

Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., & Sattler, U. (2008). *OWL 2: The Next Step for OWL*. https://doi.org/10.1016/j.websem.2008.05.001

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220. https://doi.org/10.1006/knac.1993.1008

ISO/IEC/IEEE 15288. (2023). *Systems and software engineering – System life cycle processes.* (ISO Standard No. 15288:2023). https://doi.org/10.1109/IEEESTD.2023.10123367

Kang, D., Lee, J., Choi, S., & Kim, K. (2010). An ontology-based Enterprise Architecture. *Expert Systems with Applications*, *37*(2), 1456–1464. https://doi.org/10.1016/j.eswa.2009.06.073

Kulvatunyou, B., Ivezic, N., Lee, Y., & Shin, J. (2014). An analysis of OWL-based semantic mediation approaches to enhance manufacturing service capability models. *International Journal of Computer Integrated Manufacturing*, 27(9), 803–823. https://doi.org/10.1080/0951192X.2013.834477

Ludascher, B., Gupta, A., & Martone, M. E. (2001). Model-based mediation with domain maps. *Proceedings 17th International Conference on Data Engineering*, 81–90. https://doi.org/10.1109/ICDE.2001.914816

Lütjen, M., Kreowski, H. J., Franke, M., Thoben, K. D., & Freitag, M. (2014). Model-driven logistics engineering – challenges of model and object transformation. *2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering*. Procedia Technology 15(303-312). https://doi.org/10.1016/j.protcy.2014.09.084

Malviya N, Mishra N, & Sahu S. (2011). Developing university ontology using protégé OWL tool: Process and reasoning. *International Journal of Scientific and Engineering Research*. 2(9).

Mkhinini, M. M., Labbani-Narsis, O., & Nicolle, C. (2020). Combining UML and ontology: An exploratory survey. *Computer Science Review*, 35, 100223. https://doi.org/10.1016/j.cosrev.2019.100223

Munn, K., & Smith, B. (2013). *Applied Ontology: An Introduction*.

NASA. *NASA Human Systems Integration Handbook*. (2021). https://ntrs.nasa.gov/api/citations/20210010952/downloads/HSI%20Handbook%20v2.0%20092121_FINAL%20COPY.pdf. Accessed 16 June 2024.

Noy, N. F., & McGuinness, D. L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*.

Shani, U., Jacobs, S., Wengrowicz, N., & Dori, D. (2016). *Engaging ontologies to break MBSE tools boundaries through semantic mediation*. In Proceedings of the 2016 Conference on Systems Engineering Research, Huntsville, AL, USA, March 2016.

Uschold, M., & Jasper, R. (1999). *A Framework for Understanding and Classifying Ontology Applications*. Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden, August 2, 1999.

van Ruijven, L. (2015). Ontology for Systems Engineering as a base for MBSE. *INCOSE International Symposium*, 25, 250–265. https://doi.org/10.1002/j.2334-5837.2015.00061.x

World Wide Web Consortium (W3C). (2004). *RDF 1.2 Concepts and Abstract Syntax*. Retrieved March 16, 2024, from https://www.w3.org/TR/rdf12-concepts/.