Cybersecurity and Innovative Technology Journal, Vol.2, No.2, 2024, pp. 79-97
e-ISSN 3025-6682. DOI. 10.52889/citj.v2i2.540

Ontology Development for SysML-Based Energy Harvester Design

Submitted 23 November 2024 Revised 31 December 2024 Accepted 31 December 2024

Christopher W. Green[1*]

[1]Department of Systems Engineering, Capitol Technology University, Laurel, United States

Corresponding Email: *cgreen@captechu.edu

Abstract

This study presented the development of an ontology for energy harvesting systems, leveraging the Systems Modeling Language (SysML) within a Model-Based Systems Engineering (MBSE). Energy harvesting, which transforms ambient energy into usable electrical power, has gained traction as a sustainable solution for powering low-power devices in applications such as wireless sensors and autonomous systems. This paper explored the diverse sources and components involved in energy harvesting, including transduction mechanisms, power management, and storage, to address the complexity inherent in multi-source energy systems. Using SysML profiles, this study constructs an ontology that formalizes the concepts, relationships, and constraints within the energy harvesting domain, providing a common vocabulary and promoting semantic interoperability among stakeholders. The ontology is developed systematically and structured to facilitate designing and reusing energy harvesting components, supporting increased efficiency and scalability. The results demonstrate how ontology-based design aids in reducing complexity, improving consistency, and fostering collaboration across interdisciplinary teams, offering significant implications for sustainable technology development in energy harvesting applications.

Keywords: Ontology, SysML, MBSE, Energy Harvesting

INTRODUCTION

The motivation of this work is to reduce the use of energy sources that harm the environment. This will be done by reducing the production time for the design of devices that utilize multiple alternative/ renewable energy sources to power deployed devices. The desired effect is to reduce the environmental pollution footprint by reducing the use of batteries and their subsequent disposal, contributing to the overuse of landfills. Below are the research questions that will be answered.

1) What are the benefits and challenges of using ontologies to enhance the Systems Modeling Language (SysML) modeling of energy harvesting systems?
2) What key concepts and relationships should be included in an ontology for energy harvesting systems?

Energy Harvesting

Energy harvesting transforms ambient energy into electrical energy that can be used to operate low-power devices. Power harvesting and energy scavenging are other names synonymous with energy harvesting. Typical energy sources include waste (thermal, kinetic, and electromagnetic) and renewable energy (sun, wind, and ocean). Other energies, such as those produced by industrial machinery, moving automobiles, buildings, natural sources, human activity, and organ movement, can be captured and transformed into usable electric power. Various energy harvesting techniques have been put forth at the meso, micro, and

Cybersecurity and Innovative Technology Journal, Vol.2, No.2, 2024, pp. 79-97
e-ISSN 3025-6682. DOI. 10.52889/citj.v2i2.540

nanoscale utilizing transduction mechanisms such as electromagnetic, electrostatic, piezoelectric, triboelectric, thermoelectric, and pyroelectric. (Sezer & Koç, 2021). Multi-source energy harvesters have advantages, including increased reliability and efficiency relative to the load. However, because of the various types of processing components needed for multiple sources, the complexity of these harvesters increases (Shi et al., 2023)

Several things must be considered when designing energy harvesters. On a macro level, the energy source, the type of harvester, and the intended application. When choosing the type of energy harvester, the following should be considered (Deng et al., 2019):

- Energy Conversion Technology
- Integration of Multiple Harvesters
- Power Management
- Energy Storage and Management
- System Optimization and Control

A system engineering approach can be adopted to facilitate the design of energy harvesters and manage complexity. Model-based systems Engineering emphasizes the use of models throughout the engineering lifecycle. Model-Oriented Systems Architecting (MOSA) is a methodology that emphasizes using models to drive the system architecture. A Reference Architecture, which provides a common framework for designing harvesters, can be developed using both tools. This chapter focuses on a literature search to define this development (MOSA, 2020).

A typical energy harvesting system consists of the following basic components:

- Stimulus: the ambient energy source
- Converter: Converts the ambient energy into electrical energy
- Power Electronics: The hardware and software that process the energy into a form that is useable to the rest of the system
- Storage: Rechargeable cells, batteries, and supercapacitors that store harvested energy for use
- Application: The use that consumes the harvested energy

Figure 1 depicts an energy harvesting system. Figure 2 shows the stimuli or energy sources in relation to the system.

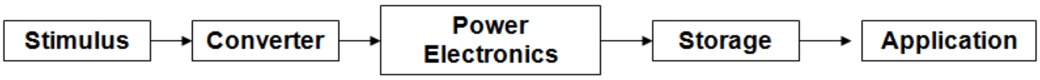


Figure 1: Energy Harvesting System

Cybersecurity and Innovative Technology Journal, Vol.2, No.2, 2024, pp. 79-97
e-ISSN 3025-6682. DOI. 10.52889/citj.v2i2.540

Depending on the type of converter, the power electronics may face the technical challenge of rectification or voltage conversion. AC-DC half-wave or full-wave rectification is necessary for converters that produce an A.C. output signal. Similarly, depending on the application for converters that produce a D.C. output signal, D.C. to D.C. conversion may be required (Erickson, 2020). Impedance matching between the converter and the electronics is a crucial design metric when optimizing energy transfer (Sodano, 2006). Advanced systems may include a microcontroller that executes power management algorithms. Storage consists of rechargeable batteries or a capacitor. Standard rechargeable batteries include lead-acid, nickel-cadmium (NiCd), nickel metal hydride (NiMH), lithium-ion (Li-ion), and lithium-ion polymer (Li-ion polymer). Each type of battery has pros and cons, so the design is application-specific (Beard, 2019).

Wireless sensors and sensor networks have been the primary motivation for developing energy harvesters. Integrating a wireless sensor with an energy harvester to power the sensor directly or to recharge the sensor's battery would significantly reduce operational costs. (Chai et al., 2020) (Shirvanimoghaddam et al., 2019).

Harvested energy for autonomous devices can be divided into five categories: radiant, mechanical, thermal, magnetic, and biochemical.

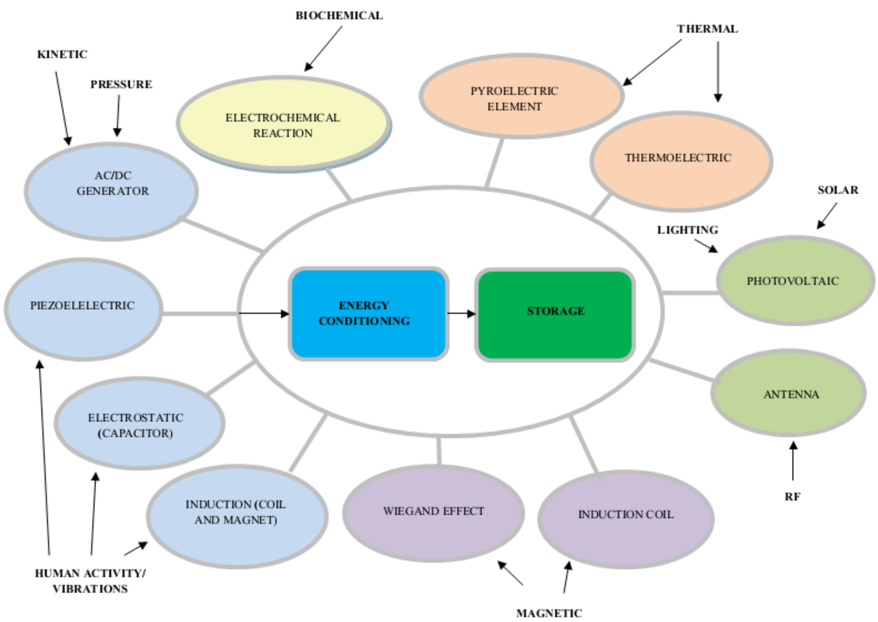


Figure 2: Comprehensive View of the Energy Harvesting System

Other parts of an energy harvesting system or harvester include the power electronics, storage, and application.

Ontology Development

Ontology modeling has a rich history that originated in philosophy, computer science, and artificial intelligence. The term "ontology" originates from philosophy, where it refers to the study of being and the categorization of existence. Early philosophical frameworks, such as Aristotle's classifications of substances, laid the groundwork for structuring knowledge systematically (Smith, 2003). In computer science, the concept of ontology modeling emerged in the 20th century as researchers sought ways to represent knowledge computationally. One of the earliest efforts was in the 1970s when artificial intelligence researchers like John McCarthy developed formal systems to represent knowledge for reasoning and problem-solving (Guarino, 1998).

The 1990s marked a significant turning point in ontology modeling, driven by the rise of knowledge management and the need for standardization in knowledge representation. During this period, methodologies like the Enterprise Ontology and the Toronto Virtual Enterprise (TOVE) ontology were developed, providing structured approaches for modeling business processes and organizational knowledge (Uschold & Gruninger, 1996). The World Wide Web Consortium (W3C) played a pivotal role in advancing ontology modeling for the semantic web, introducing the Resource Description Framework (RDF) and the Web Ontology Language (OWL) in the early 2000s to enable machine-readable and interoperable web data (Berners-Lee et al., 2001).

Foundational ontologies such as the Suggested Upper Merged Ontology (SUMO) and Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) emerged during this period, providing high-level frameworks for structuring knowledge across domains (Niles & Pease, 2001; Masolo et al., 2003). These ontologies served as a starting point for creating domain-specific models in fields like medicine, finance, and engineering. Description Logics (DL), a family of knowledge representation languages, also became a formal foundation for ontology modeling, influencing the development of OWL and related tools (Baader et al., 2007).

Today, ontology modeling integrates insights from multiple disciplines, including linguistics, cognitive science, and social sciences. Researchers such as Ian Horrocks and organizations like the Ontology Engineering Group have contributed significantly to advancing tools and methodologies for creating and maintaining ontologies. These developments have broadened the applications of ontology modeling, enabling innovations in artificial intelligence, data integration, and knowledge management (Horrocks et al., 2003). The continued evolution of ontology modeling underscores its foundational role in structuring and leveraging knowledge in diverse domains.

Ontology within the context of this work falls within the domain of information science. (Noy & McGuinness, 2001) define an ontology as "ontology is a formal, explicit description of concepts in a domain of discourse (classes (sometimes called concepts)), properties of each concept describing various features and attributes of the concept (slots (sometimes called roles or properties)), and restrictions on slots (facets (sometimes called role restrictions)) ."The main purpose of an ontology is to define a common vocabulary, concepts, and relationships within a domain. The rationale for developing an ontology is as follows:

- To share a common structure of domain knowledge
- To enable the reuse of domain knowledge
- To make assumptions explicit
- To analyze the domain knowledge
- To separate domain knowledge from operational knowledge

Ontologies are used in Systems Engineering to improve knowledge management by establishing well-defined concepts, relationships, and terminology within domains, thus enabling better interoperability, knowledge sharing, and reusability across systems engineering tasks (Yang et al., 2019). Ontology-based engineering leverages ontologies—structured representations of domain knowledge—to capture explicit and implicit information from experts and historical data. This knowledge is used to generate automated design alternatives for conceptual manufacturing processes. Arista et al. present a framework for enhancing aircraft manufacturing design through Ontology-Based Engineering (OBE)(Arista et al., 2022). A reference model can be derived from an ontology to capture the high-level structure and relationships within the domain (Roy et al., 2024). Domain-specific ontologies such as Ontology Web Language (OWL) can be integrated with MBSE tools for consistency checks and reasoning tasks. These ontologies provides a semantic layer that MBSE tools often lack (Medinacelli et al., 2023). SysML and Semantic Web Technologies can be used in a framework to model Distributed Automation Systems (DASs) (Wang et al., 2023). Combining metamodels (like SysML and UML) with project-specific ontologies, a hybrid approach improves Model-Based Systems Engineering (MBSE) by creating easier models for non-technical team members to understand and validate. Ernadote proposes a hybrid approach to improve Model-Based Systems Engineering (MBSE) by combining metamodels (like SysML and UML) with project-specific ontologies. This approach aims to enhance collaboration among diverse stakeholders by creating easier models for non-technical team members to understand and validate

(Ernadote, D., 2015). The use of ontologies in MBSE can help models remain relevant and functional despite changes in modeling tools or languages (Shani, U., 2017). Ontologies also allow for enhanced interoperability and resilience in MBSE models, providing a foundation for scalable, AI-enabled MBSE applications that facilitate collaborative engineering and lifecycle data management (Lu et al., 2022).

An ontology is crucial for defining and managing architectures in various domains because it provides a formal and structured representation of the concepts, relationships, and constraints within that domain. Below are some reasons why it is a good idea to use an ontology in development of an architecture:

- Common Knowledge: An ontology creates a common vocabulary and understanding among the parties constructing the architecture. Defining standardized terminology, concepts, and relationships ensures clarity and consistency in communication.
- Knowledge Management: Domain-specific information, appropriate rules, and architectural constraints are captured and arranged in an ontology, a knowledge repository.
- Semantic Interoperability: Systems, parts, and technologies from several disciplines are frequently integrated into architectures. An ontology facilitates semantic interoperability by formalizing the semantics and relationships between architectural elements. It improves consistency and coherence throughout the design by facilitating smooth integration and information sharing amongst stakeholders, tools, and artifacts.
- Reusability: An ontology aids in the reuse of architectural knowledge and components. It helps architects create new systems by formalizing domain concepts, patterns, and best practices to use preexisting structures and design principles.

Ontology is essential to architecture because it gives formalized architectural knowledge to a home, encourages cooperation and interoperability, supports knowledge management, encourages reuse and traceability, and allows flexibility and scalability in creating and analyzing architectural designs. It facilitates efficient communication, decision-making, and system analysis throughout the design lifecycle as a fundamental component supporting the success of architecture initiatives.

METHOD

Approach

The study used a mixed-methods reseach design, which combine qualitative and quantitative methods. The literature search method will be a narrative review or a traditional literature review, as well as a qualitative summary and synthesis of existing literature on a particular topic. It does not require a formal quality or risk of bias assessment. It does not follow a strict methodology or adhere to predefined criteria. A narrative review provides a more flexible and interpretive approach to summarizing research findings, which can inform practice, policy, and future research directions (Cronin et al., 2008) (Green B. et al., 2006) (Levy & Ellis, 2006).

The literature review will be evaluated from both perspectives because an ontology will incorporate both data types. The ontology structure, which will specify the concepts and language, will be derived from qualitative data, and the fields within the structure, such as attributes, will be derived from quantitative data.

Ontology Process

Ontologies can be represented using RDF (Resource Description Framework) and OWL (Web Ontology Language). These languages can describe the parts of an ontology, namely the classes and relations between them. A systematic process should be followed to develop useful ontologies and is listed below (Lamy, 2020) (Devedvic et al., 2009):

- Step 1. Determine the domain and scope of the ontology
- Step 2. Consider reusing existing ontologies
- Step 3. Enumerate important terms in the ontology
- Step 4. Define the classes and the class hierarchy
- Step 5. Define the properties of classes—slots
- Step 6. Define the facets of the slots
- Step 7. Create instances

Ontology Diagram Elements

Profile

A profile is a mechanism to extend the language by defining specialized sets of stereotypes, constraints, and modeling conventions tailored to specific domains, industries, or applications. Profiles introduce domain-specific concepts, properties, and relationships into the system model. Below are some key profile aspects (Hemmert & Schweiger, 2022).

- Customization: Profiles allow the SysML language to be extended and customized to meet domain-specific modeling needs. Profiles are specific stereotypes, tagged values, constraints, and modeling rules that reflect system components' distinct behaviors, relationships, and attributes.
- Extension of Stereotypes: Profiles can define or reinforce new stereotypes. Stereotypes specify particular kinds or classifications of model elements with traits, actions, and meanings tailored to the demands of the specific domain.
- Semantic Annotation: Profiles add semantic information to model elements beyond what

the normal SysML language constructs can capture. Profiles can include descriptive tags, limitations, or recommendations that specify the intended use or meaning of the stereotyped elements within the system model.

- Reuse and Interoperability: Profiles help promote reuse and interoperability by standardizing the representation of domain-specific concepts and components across many models and projects.

Stereotypes

Stereotypes are an element of a profile. Figure 3 depicts a physical view of stereotypes, showing the structure of a diagram. Here are key aspects of stereotypes in SysML (Hemmert & Schweiger, 2022):

- Extension of Model Elements: Stereotypes can provide new characteristics or behaviors and expand the semantics of current SysML model components, such as requirements, activities, blocks, and relationships.
- Domain-Specific Modeling: Stereotypes adapt the SysML language to certain fields, applications, or businesses, enabling domain-specific modeling. To facilitate the construction of specialized system models that effectively represent the domain context and solve its particular challenges and objectives, modelers might define stereotypes that reflect the distinctive qualities, requirements, and limitations of their domain.

Classes

A class is a basic idea that describes a system's behavior and structure of parts, objects, and subsystems. Classes in object-oriented programming (OOP) languages, such as Java, Python, or C++, are comparable to those in SysML. Below are some key aspects of SysML classes.

- Structure: A class represents a system component, such as a part, object, or subsystem. It defines the properties, attributes, and relationships associated with instances of that component type.
- Attributes: Describes the characteristics or properties. These represent the data associated with class instances and may include properties such as name, type, value, and constraints.
- Operations: Defines the actions or functions associated with instances of the class.
- Relationships: Classes can be related to other classes through various relationships, such as associations, aggregations, compositions, and generalizations.
- Inheritance: A concept where one class inherits properties, attributes, and operations from another.
- Constraints: Specify conditions or rules that class instances must satisfy.

Cybersecurity and Innovative Technology Journal, Vol.2, No.2, 2024, pp. 79-97
e-ISSN 3025-6682. DOI. 10.52889/citj.v2i2.540

Enumeration

An enumeration is a modeling construct that defines a fixed set of named values or literals within a domain-specific context. Enumerations are a set of symbolic names (often referred to as "enumerators" or "constants") assigned to represent distinct values. They define properties, attributes, parameters, and other model elements with a discrete set of possible values. Enumerators provide a structured and standardized way to represent categorical data and constraints within the system model.

Relationships

Relationships define connections, associations, dependencies, or interactions between model elements within a system model. Relationships are essential for representing system components' structural, behavioral, and conceptual dependencies. The six main types of relationships between classes are association, generalization, dependency, flow, aggregation and composition, and port and connector. Below are descriptions of these relationships.

- Association: An association represents a bi-directional relationship between two or more model elements, typically between blocks or classes.
- Generalization: Generalization represents an inheritance relationship between two model elements, typically between blocks or classes.
- Dependency: Dependency represents a relationship between two model elements where a change in one element may impact or affect another.
- Flow: Flow represents the transfer of data, energy, or material between model elements within the system.
- Aggregation and Composition: Aggregation and composition represent part-whole relationships between model elements, typically between blocks or parts.
- Port and Connector: Ports define the interfaces of blocks or components, while connectors represent the connections between ports that enable communication and interaction between elements.

Relationships capture the structural, behavioral, and conceptual dependencies between model elements within a system model. They enable engineers and designers to represent the complex interactions and dependencies between system components, facilitating the analysis, design, and documentation of systems in a structured and organized manner. The arrows for the six relationships are shown in Figure 4 (Friedenthal et al., 2014).

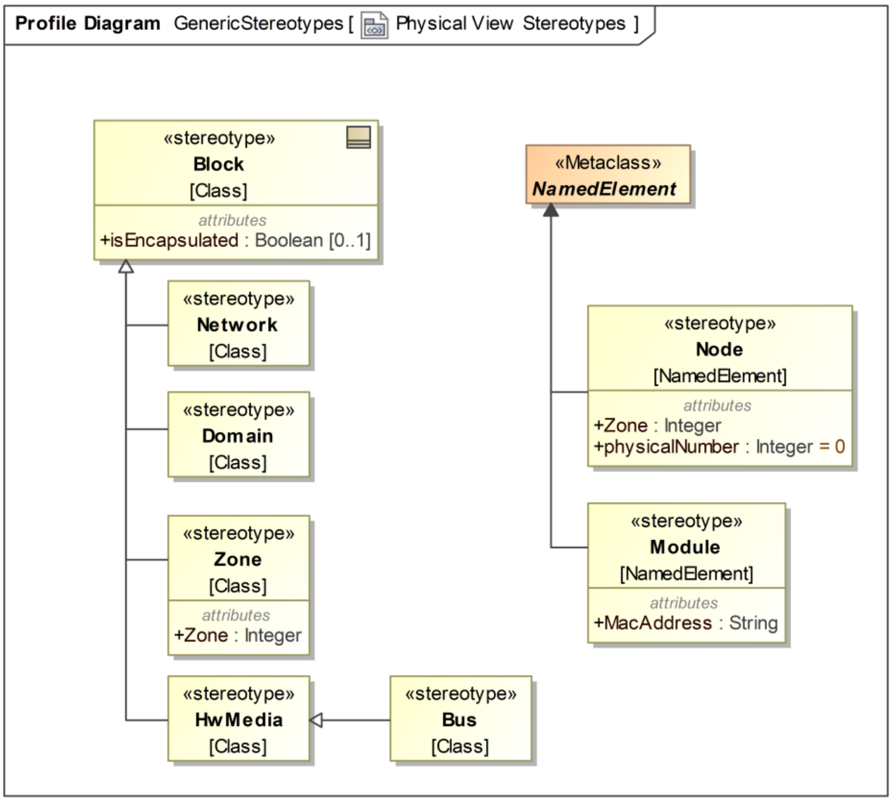


Figure 3 Physical View Stereotypes

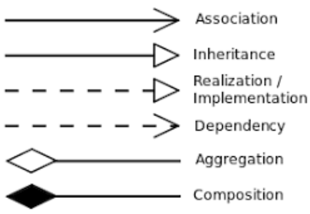


Figure 4 Arrows for the Relationships

RESULTS AND DISCUSSION

Results

This section's scenario involves an engineering team designing an energy harvesting system for an electric car. The goal is to extend the car's driving range. The team plans to incorporate a multi-source energy harvester that can capture waste thermal and vibrational energy from various sources within the vehicle.

The introduction describes the components of an energy harvesting system that were used in the ontology. Figures 5, Figure 6, Figure 7, Figure 8, Figure 9, Figure, 10, Figure 11, Figure 12, Figure 13, and Figure 14 depict the ontology, starting with the system and progressing through the subsystems with the appropriate classes and enumerations.

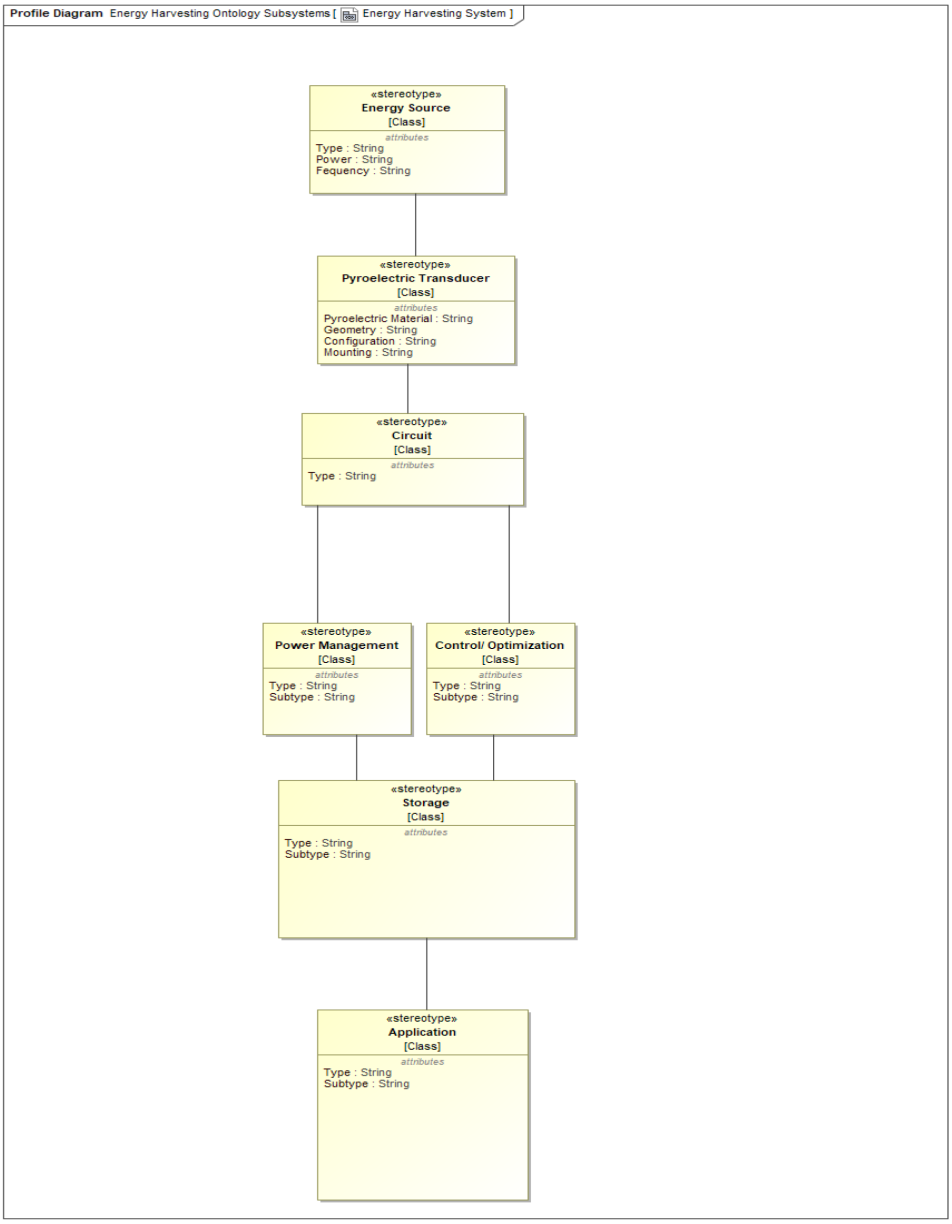


Figure 5 Energy Harvesting System Ontology Diagram

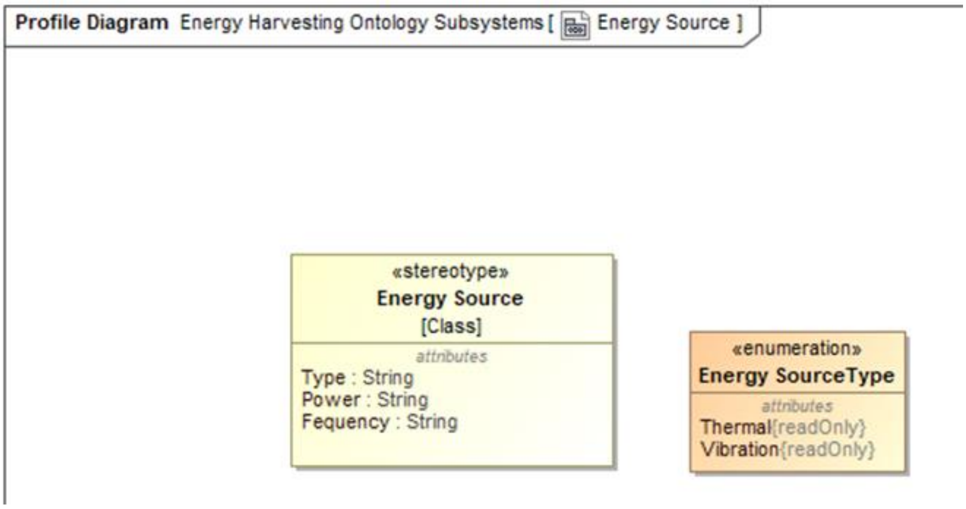


Figure 6 Energy Source Ontology Diagram

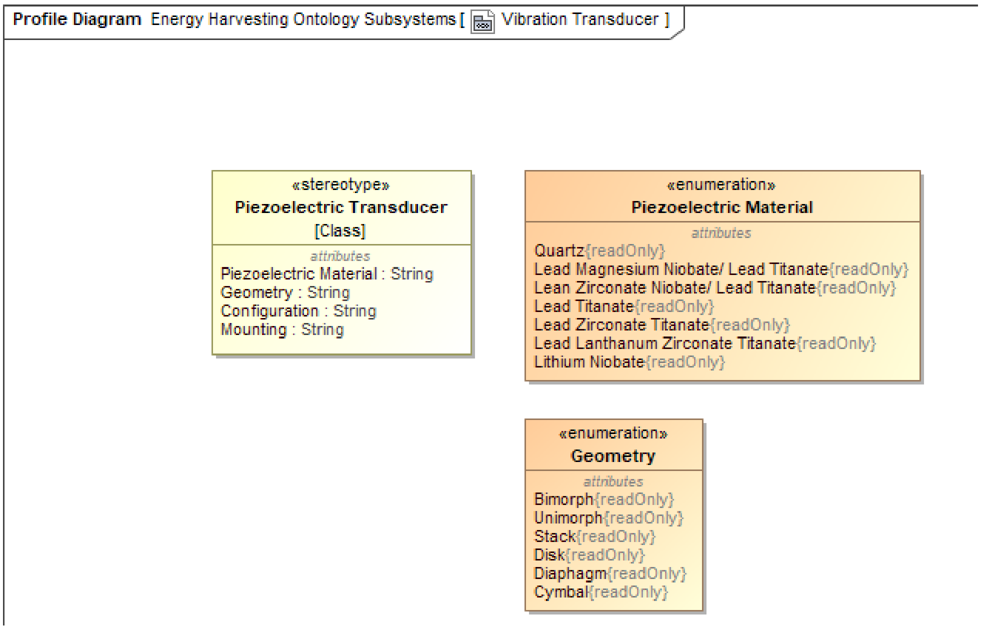


Figure 7 Vibration Transducer Ontology Diagram

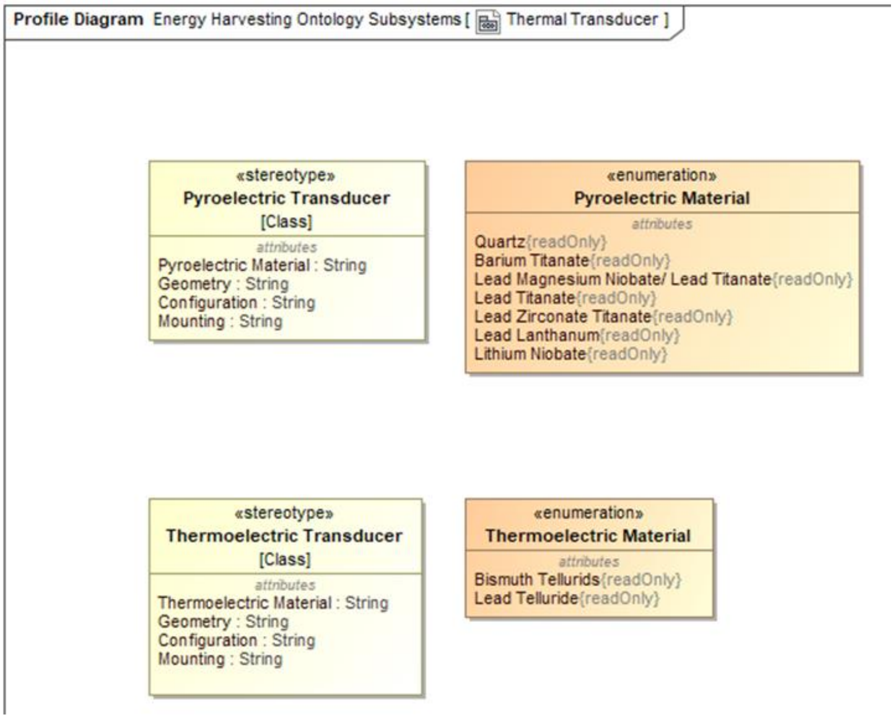


Figure 8 Thermal Transducer Ontology Diagram

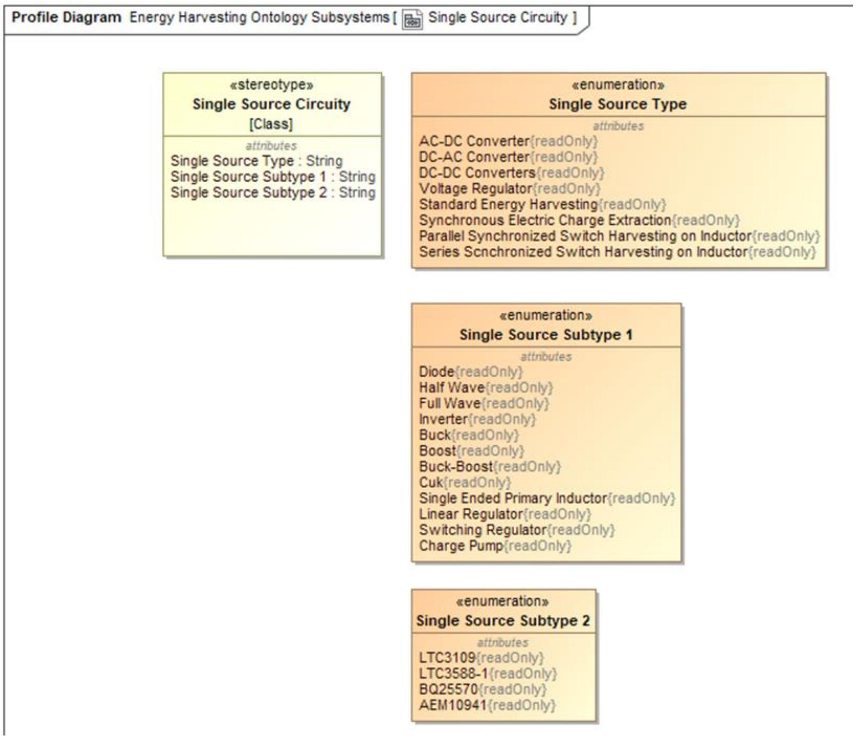


Figure 9. Single Source Circuitry Ontology Diagram

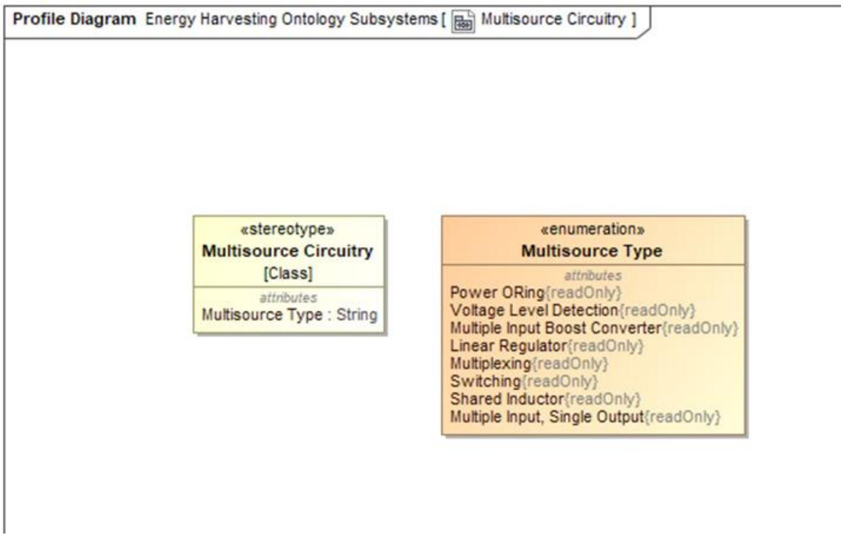


Figure 10. Multi-source Circuitry Ontology Diagram

Cybersecurity and Innovative Technology Journal, Vol.2, No.2, 2024, pp. 79-97
e-ISSN 3025-6682. DOI. 10.52889/citj.v2i2.540

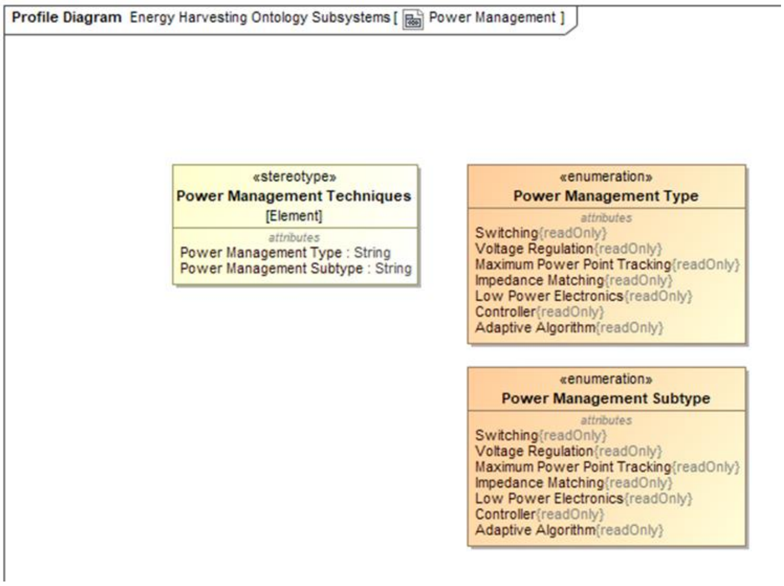


Figure 11. Power Management Ontology Diagram

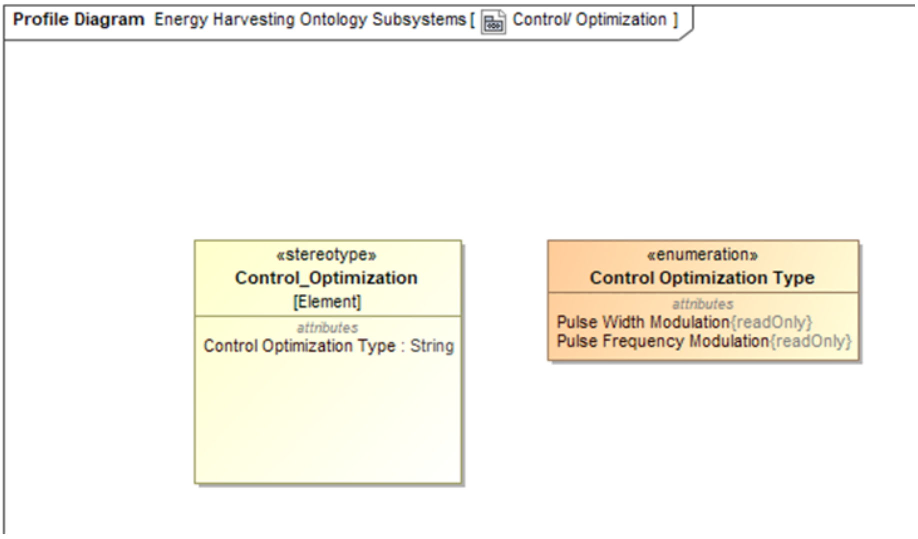
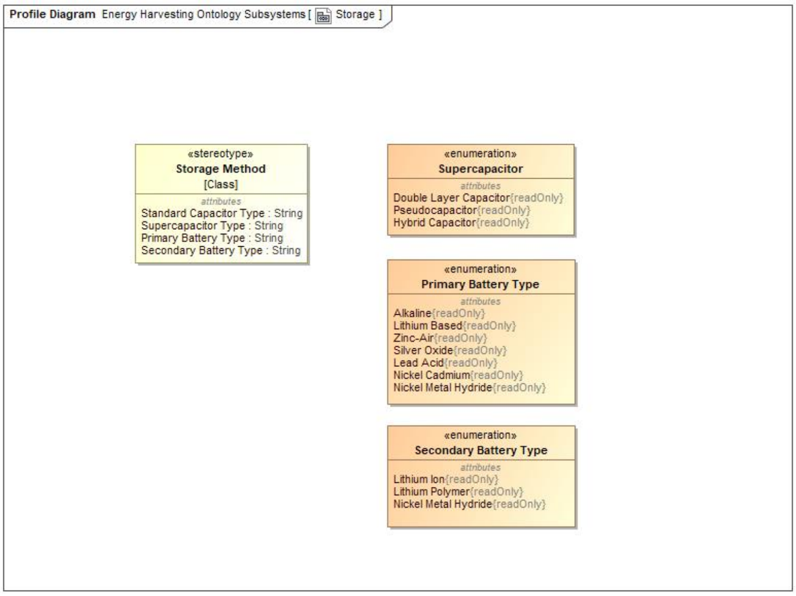


Figure 12. Control/ Optimization Ontology Diagram



Cybersecurity and Innovative Technology Journal, Vol.2, No.2, 2024, pp. 79-97
e-ISSN 3025-6682. DOI. 10.52889/citj.v2i2.540

Figure 13. Storage Ontology Diagram

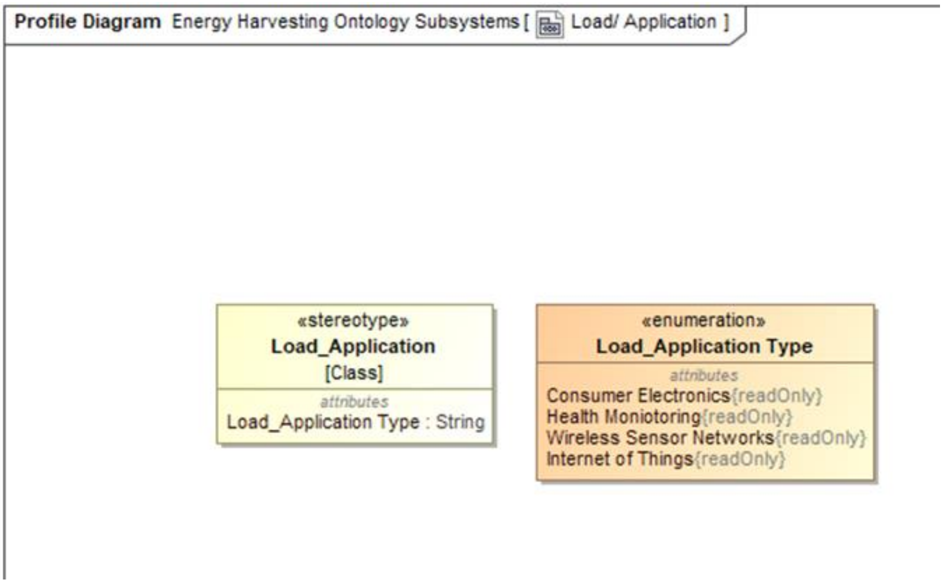


Figure 14. Load Ontology Diagram

Discussion

This section will address the research questions from the introduction section. This structured ontology aids in developing modular, scalable, and optimized SysML models for energy harvesting systems.

Research Question 1. Benefits and Challenges of Using Ontologies to Enhance SysML Modeling of Energy Harvesting Systems.

Benefits:

- Standardized Semantics and Syntax: Ontologies provide a consistent framework for describing system components and their interactions, enhancing clarity in SysML models.
- Improved Communication: They bridge the gap between stakeholders by establishing a common vocabulary.
- Enhanced Interoperability: Ontologies support integration across various tools and platforms, enabling seamless data sharing and reuse.
- System Complexity Management: Ontologies facilitate the management and navigation of system complexity by formalizing the relationships and hierarchies within energy harvesting systems.
- Support for Multi-Source Systems: Ontologies facilitate the representation of diverse energy sources, converters, and management strategies, which are essential in multi-source energy harvesting.

Challenges:

- Development Complexity: Requires significant effort to ensure it captures all relevant aspects of energy harvesting systems.
- Integration with Existing Tools: Ensuring compatibility between ontology frameworks and SysML tools can be technically challenging.
- Dynamic Nature of Energy Systems: Energy harvesting systems evolve, necessitating regular updates to the ontology, which can be resource-intensive.
- Learning Curve: Training may be required to use and apply ontologies within SysML modeling effectively.

Research Question 2. Key Concepts and Relationships to Include in an Ontology for Energy Harvesting Systems

An ontology for energy harvesting systems should include the following key concepts and their relationships:

- Energy Sources:
  - Types: Solar, wind, vibration, thermal, electromagnetic, biological.
  - Relationships: Energy sources link to specific converters or transducers.
- Energy Converters/Transducers:
  - Types: Piezoelectric, thermoelectric, electromagnetic, photovoltaic.
  - Relationships: Converters are associated with specific energy sources and power electronics.
- Power Electronics:
  - Components: Rectifiers, converters (buck, boost, buck-boost), charge pumps.
  - Relationships: Power electronics link energy converters to storage or directly to applications.
- Energy Storage:
  - Types: Batteries, supercapacitors.
  - Attributes: Capacity, efficiency, discharge rate.
  - Relationships: Storage systems interact with power electronics and applications.
- Applications:
  - Examples: IoT devices, wireless sensor networks, consumer electronics.
  - Relationships: Applications are powered by energy harvested through the system.
- Control and Optimization:
  - Techniques: Maximum Power Point Tracking (MPPT), adaptive algorithms.
  - Relationships: Control strategies optimize interactions between components.
- System Metrics:

- Measures: Efficiency, power density, reliability.
- Relationships: Metrics evaluate the performance of the system and its components.

CONCLUSION

Creating an ontology with SysML profiles involves defining a set of stereotypes, constraints, and modeling conventions within a SysML profile to represent domain-specific concepts, relationships, and constraints. Ontologies capture a domain's semantics by formalizing the concepts, relationships, and constraints that govern the domain and its components. This study defines an ontology in the context of energy harvesting. The domain language was described in the literature. A process was used for ontology development, and diagrams were created using Cameo Systems Modeler. The research questions were answered in the execution of this study.

REFERENCES

Arista, R., Zheng, X., Lu, J., & Mas, F. (2023). An Ontology-Based Engineering System To Support Aircraft Manufacturing System Design. Journal of Manufacturing Systems, 68, 270-288. https://doi.org/10.1016/j.jmsy.2023.02.012

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (2007). The Description Logic Handbook: Theory, Implementation, And Applications. Cambridge University Press.

Beard, K. W. (2019). Linden’s Handbook of Batteries (5th ed.). McGraw-Hill.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. Scientific American, 284(5), 28-37.

Chai, S., Wang, Z., Zhang, B., Cui, L., & Chai, R. (2020). Wireless Sensor Networks (1st ed.). Springer Nature Singapore. https://doi.org/10.1007/978-981-15-5757-6

Chang, C., Xiao, G., & Zhang, Y. (2021). Ontology for Systems Engineering Technical Processes. In Complex Systems Design & Management. Springer, Cham. https://doi.org/10.1007/978-3-030-73539-5_26

Deng, F., Yue, X., Fan, X., Guan, S., Xu, Y., & Chen, J. (2019). Multi-source Energy Harvesting System for a Wireless Sensor Network Node in the Field Environment. IEEE Internet of Things Journal, 6(1), 918–927. https://doi.org/10.1109/JIOT.2018.2865431

Devedvic, V., Djuric, D., & Gasevic, D. (2009). Model Driven Engineering and Ontology Development (2nd ed.). Springer-Verlag. https://doi.org/10.1007/978-3-642-00282-3

Duprez, J., & Ernadote, D. (2020). Towards A Semantic Approach Of MBSE Frameworks Specification. INCOSE International Symposium, 30(1), 1416-1430. https://doi.org/10.1002/j.2334-5837.2020.00794.x

Ernadote, D. (2015). An Ontology Mindset For System Engineering. IEEE International Symposium on Systems Engineering (ISSE), Rome, Italy, pp. 454-460. https://doi.org/10.1109/SysEng.2015.7302797

Ernadote, D. (2017). Ontology-Based Pattern for System Engineering. ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), Austin, TX, USA, pp. 248-258. https://doi.org/10.1109/MODELS.2017.4

Friedenthal, S., Moore, A., & Steiner, R. (2012). A Practical Guide to SysML: The System Modeling Language (2nd ed.). Morgan Kaufmann.

Gašević, D., Djurić, D., & Devedzic, V. (2006). Model Driven Architecture and Ontology Development. Springer-Verlag.

Guarino, N. (1998). Formal ontology in information systems. In Proceedings of FOIS '98 (pp. 3-15). IOS Press.

Hemmert, A., & Schweiger, A. (2022). Development of a SysML Profile for Network Configurations in Safety-Critical Systems. Modellierung 2022 Satellite Events. https://doi.org/10.18420/modellierung2022ws-007

Horrocks, I., Patel-Schneider, P. F., & van Harmelen, F. (2003). From SHIQ and RDF to OWL: The Making Of A Web Ontology Language. Web Semantics: Science, Services, and Agents on the World Wide Web, 1(1), 7-26.

Lu, J., Ma, J., , Zheng, X., Wang, G., Li, H. Kiritsis, D. (2022). Design Ontology Supporting Model-Based Systems Engineering Formalisms. IEEE Systems Journal, vol. 16, no. 4, pp. 5465-5476, Dec. 2022, doi: 10.1109/JSYST.2021.3106195.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., & Oltramari, A. (2003). WonderWeb deliverable D18: Ontology library (final report). ISTC-CNR.

Medinacelli, L. P., Noyrit, F., & Mraidha, C. (2023). A Methodology For Knowledge Integration And Acquisition In Model-Based Systems Engineering. CEA Scientific production in open Access. https://doi.org/10.5220/0012233900003598

Niles, I., & Pease, A. (2001). Towards a standard upper ontology. In Proceedings Of The International Conference On Formal Ontology In Information Systems (pp. 2-9). ACM.

Noy, N., & McGuinness, D. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University. https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Roy, S., Huang, Y., Bekdache, D., Fung, T-Y., Beck, B., Guariniello, C., DeLaurentis, D.. (2024). From Ontology To System Architecture: An MBSE Approach Toward The Realization Of Urban Air Mobility. 34th Congress of the International Council of the Aeronautical Sciences (ICAS). September 9-13, 2024, Florence, Italy. https://www.icas.org/ICAS_ARCHIVE/ICAS2024/data/papers/ICAS2024_0554_paper.pdf

Sezer, N., & Koç, M. (2021). A Comprehensive Review On The State-Of-The-Art Of Piezoelectric Energy Harvesting. Nano Energy, 80, 105567. https://doi.org/10.1016/j.nanoen.2020.105567

Shani, U. (2017). Can Ontologies Prevent MBSE Models From Becoming Obsolete? IEEE International Systems Conference (SysCon), Montreal, QC, Canada, pp. 1-8. https://doi.org/10.1109/SYSCON.2017.7934726

Shi, G., Chang, J., Xia, Y., Wang, X., Xia, H., & Li, Q. (2023). A Multi-source Collaborative Energy Extraction Circuit for Vibration, Ambient Light, and Thermal Energy With MPPT and Single Inductor. IEEE Transactions on Industrial Electronics, 70(6), 5819–5829. https://doi.org/10.1109/TIE.2022.3196395

Shirvanimoghaddam, M., Shirvanimoghaddam, K., Abolhasani, M. M., Farhangi, M., Zahiri Barsari, V., Liu, H., Dohler, M., & Naebe, M. (2019). Towards a Green and Self-Powered Internet of Things Using Piezoelectric Energy Harvesting. IEEE Access, 7, 94533–94556. https://doi.org/10.1109/ACCESS.2019.2928523

Smith, B. (2003). Ontology. In L. Floridi (Ed.), Blackwell guide to the philosophy of computing and information (pp. 155-166). Blackwell Publishing.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods, and applications. The Knowledge Engineering Review, 11(2), 93-136.

van Ruijven, L. C. (2015). Ontology for Systems Engineering as a base for MBSE. INCOSE International Symposium, 25(1), 250-265. https://doi.org/10.1002/j.2334-5837.2015.00061.x

Vaneman, W. K. (2018). Evolving Model-Based Systems Engineering Ontologies and Structures. INCOSE International Symposium, 28(1), 1027-1036. https://doi.org/10.1002/j.2334-5837.2018.00531.x

Wang, D., Li, X., Gu, B., Cao, Y.,m Liu, Y.. (2023). An Architecture Modeling Framework for Distributed Automation Systems Using SysML and Semantic Web Technologies. 2023 6th International Conference on Mechatronics, Robotics and Automation (ICMRA) (Xiamen, China, 2023, pp. 191-200, doi: 10.1109/ICMRA59796.2023.10708094.

Yang, L., Cormican, K., & Yu, M. (2019). Ontology-Based Systems Engineering: A State-Of-The-Art Review. Computers in Industry, 111, 148-171. https://doi.org/10.1016/j.compind.2019.05.003

Zhang, J., & Yang, S. (2024). Recommendations for the Model-Based Systems Engineering Modeling Process Based on the SysML Model and Domain Knowledge. Applied Sciences, 14, 4010. https://doi.org/10.3390/app14104010