Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
ASEE
Paper ID #46869

An Ontology-Based Reasoner in Aerospace Engineering Education

Waterloo Tsutsui, Purdue University at West Lafayette (COE)

Dr. Waterloo Tsutsui is a Senior Research Associate in the School of Aeronautics and Astronautics at Purdue University, Indiana. Tsutsui received his Ph.D. in aeronautics and astronautics from Purdue University in 2017. Before joining Purdue, Tsutsui practiced engineering in the automotive industry for more than a decade, culminating as a Senior Mechanical Engineer focused on research and development of lithium-ion battery systems for electric vehicles. His research interests include systems engineering, structures and materials, product design and advanced manufacturing, and engineering education. Tsutsui is the recipient of the 2023 Engineering Education Excellence Award from the National Society of Professional Engineers (NSPE).

Yinchien Huang, Purdue University at West Lafayette (COE)

Yinchien Huang is a Ph.D. candidate in the School of Aeronautics and Astronautics at Purdue University. His research focuses on System of Systems (SoS) engineering, AI-assisted system engineering, ontology, Urban Air Mobility (UAM), and space systems. He is developing an LLM-based tool to support SoS decision-making and data sources integration, enhancing the efficiency and effectiveness of complex system interactions.

Derek Carpenter, Purdue University

Derek Carpenter is a PhD student researcher at Purdue University. His main research interests are systems engineering, hypersonics, and applications to MBSE. Derek received his Bachelor of Science in Aeronautical and Astronautical Engineering from Purdue in 2021, followed by his Master of Science in Aeronautics and Astronautics from Purdue in 2022.

Dr. Jitesh H Panchal, Purdue University at West Lafayette (PWL) (COE)

Dr. Jitesh H. Panchal is a Professor and Associate Head of Mechanical Engineering at Purdue University. He received his BTech (2000) from the Indian Institute of Technology (IIT) Guwahati, and MS (2003) and PhD (2005) in Mechanical Engineering from Georgia Institute of Technology. Dr. Panchal's research interests are in (1) design at the interface of social and physical phenomena, (2) computational methods and tools for digital engineering, and (3) secure design and manufacturing. He is a recipient of CAREER award from the National Science Foundation (NSF); Young Engineer Award, Guest Associate Editor Award, and three best paper awards from ASME; and was recognized by the B.F.S. Schaefer Outstanding Young Faculty Scholar Award, the Ruth and Joel Spira Award, and as one of the Most Impactful Faculty Inventors at Purdue University. He received the Distinguished Alumni award from IIT Guwahati. He is a co-author of two books and has co-edited one book on engineering systems design. He has served on the editorial board of international journals including ASME Journal of Mechanical Design, ASME Journal of Computing and Information Science in Engineering.

Daniel DeLaurentis, Purdue University at West Lafayette (COE)

Dr. Daniel DeLaurentis is Vice President for Discovery Park District (DPD) Institutes and the Bruce Reese professor of Aeronautics & Astronautics at Purdue University. He directs the Center for Integrated Systems in Aerospace (CISA) researching modeling, design optimization and system engineering methods for aerospace systems and systems-of-systems, including urban and regional Advanced Aerial Mobility and hypersonic systems. He is a Senior Research Fellow at the Krach Institute for Tech Diplomacy at Purdue. DeLaurentis served as Chief Scientist of the U.S. DoD's Systems Engineering Research Center (SERC) UARC from 2019-2023. He is an elected FELLOW of the American Institute of Aeronautics & Astronautics (AIAA) and the International Council on Systems Engineering (INCOSE).

An Ontology-Based Reasoner in Aerospace Engineering Education

Abstract

Building on our previous research on bottom-up (student-led) and top-down (instructor-led) approaches in aerospace engineering education, this paper presents an enhanced ontology-based reasoner that evaluates two distinct methodologies: logical consequences and word embeddings. The framework examines logical consequences' structured and rule-based query capabilities alongside word embeddings' natural language processing abilities as paths toward creating comprehensive educational tools. Our implementation demonstrates how these complementary approaches enhance educational outcomes: students benefit from personalized learning pathways and clear prerequisite relationships, while instructors gain tools for curriculum optimization and adaptation to emerging technologies. Through representative use cases, we show how these distinct approaches provide robust frameworks that balance precise, logical reasoning with flexible natural language understanding, ultimately advancing aerospace engineering education by serving both student and instructor needs.

1 Introduction

The philosophical wisdom of Aristotle, "the whole is greater than the sum of its parts," captures how true value emerges from the interactions between components, not merely their individual contributions [1]. Modern universities embody this principle as they stand at the forefront of exploring and integrating new technologies. In the early- and mid-1990s, as we started using the Internet, universities played a crucial role in adopting Internet technology like the Internet browser development of ViolaWWW and NCSA Mosaic by the University of California, Berkeley and the University of Illinois Urbana-Champaign in 1992 and 1993, respectively [2, 3]. Fast forwarding 30 years, transitioning through Web 2.0 (e.g., social networks, blogs, and video sharing) [4, 5, 6] in the 2000s and machine learning breakthroughs (e.g., deep learning [7], neural networks [8], and reinforcement learning [9]) in the 2010s, universities are now at the forefront of exploring and implementing artificial intelligence (AI). This tradition of technological leadership in academia now extends to ontology-based knowledge systems that can transform engineering education.

In this rapidly evolving landscape, engineering education faces challenges in the digital age's continuous development [10], particularly in complex fields such as aerospace engineering, where theoretical knowledge and practical applications must be effectively integrated [11]. The emergence of AI presents new opportunities to enhance educational frameworks and methodologies. One of the prominent AI examples is natural language processing via large language models (LLMs) using word embeddings [12].

To explore the integration of ontology with both logical consequences and word embeddings in creating an ontology-based reasoner, we present this research paper in the following order. We begin by reviewing existing knowledge on ontology in engineering education, followed by analyses of both student-centered and instructor-centered methodological approaches. Next, we present a theoretical framework that unifies ontologies with logical consequences (Path 1) and with word embeddings (Path 2). We then illustrate and discuss this framework through a practical use case in aerospace engineering education, demonstrating our ontology-based reasoner's application in student-centered course planning and instructor-centered curriculum development and advising. In this paper, we adopt two key perspectives, as illustrated in Fig. 1: a bottom-up approach focusing on student-centered aspects and a top-down approach addressing instructor-centered aspects. Throughout our discussion, the term "instructor" encompasses both instructors and administrators who oversee coursework.

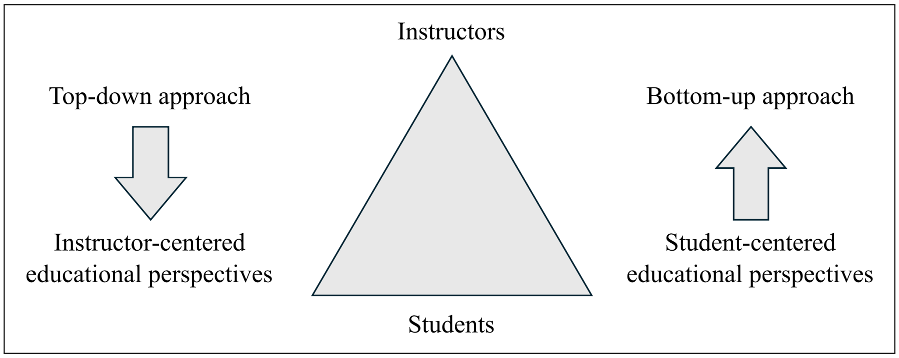

Figure 1: Educational Perspectives: Student-centered (Bottom-up) and Instructor-centered (Top-down)

2 Background

2.1 Foundations of Ontological Reasoning

Ontologies [13, 14], logical consequences [15, 16], and word embeddings [17, 18] share fundamental connections in AI, and combining these tools offers promising potential for advancing engineering education. Ontological reasoning, a comprehensive approach to deriving knowledge from structured representations, plays a crucial role [19]. Within the broader framework of ontological reasoning, logical consequences and word embeddings serve as fundamental mechanisms. We can perform various types of knowledge derivation through ontological reasoning, including classification, hierarchical relationships, and consistency checking. That is, combining the structured understanding of ontologies with the logical consequences (i.e., SPARQL, which will be covered later) and word embeddings (i.e., the foundation for LLMs) enables us to create powerful tools for educational applications, thereby enabling the systematic derivation of new knowledge from existing ontological structures. These foundational concepts form the basis for our educational framework.

2.2 Ontology in Engineering Education

Educational ontology development can employ two distinct methodologies: the top-down approach starting with general concepts and specializing downward, and the bottom-up approach beginning with specifics and organizing upward [20, 21]. The use of ontology provides a framework enabling classification (e.g., alpha, beta, and gamma levels), consistency checks, and gap identification in Aerospace systems engineering [22, 23], supporting more structured and reliable processes in systems design and analysis [24, 25]. Similarly, structured visualization frameworks in engineering education have demonstrated effectiveness in helping students understand complex physical concepts such as deformation and stress in aerospace structures [26].

Since the fundamental idea of ontology is to define relationships between entities in the form of "A is a B," the ontology-based approach facilitates the modeling and analysis of relationships among various systems. Consequently, this approach can integrate concepts from different academic disciplines. For instance, aerospace engineering can combine areas such as aerodynamics, autonomy and control, propulsion, structures, and materials. As a result, the ontology-based approach fosters cross-disciplinary learning for students and enhances teaching experiences for instructors. This approach complements previous educational frameworks in aerospace engineering that use interactive visualization technologies, which have demonstrated effectiveness in helping students understand complex physical concepts in structural mechanics through both virtual and hands-on experiences [27].

This ontology-based reasoner builds upon previous system-of-systems (SoS)-inspired frameworks in aerospace engineering education. While SoS approaches use Definition, Abstraction, and Implementation (DAI) process [28] to solve complex problems, our ontology-based reasoner similarly provides structured pathways through complex engineering knowledge.

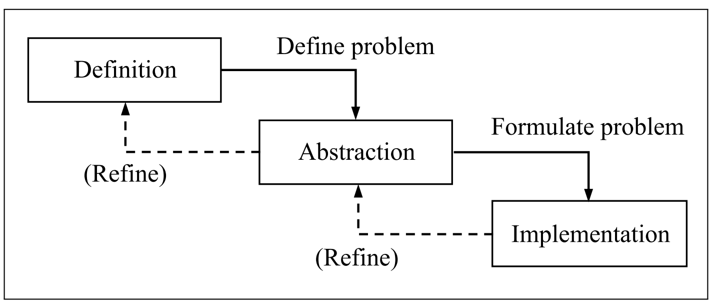

Figure 2: SoS DAI Process

2.3 Educational Framework Integration

The Backward Course Design Model focuses on learning outcomes rather than teaching methods, making it ideal for serving both instructor and student needs [29, 30, 31]. The model works in the following steps: first, setting the learning goals, then deciding how to measure success, and finally, planning the teaching activities. This approach works well with how we organize course content using ontologies. For example, the learning goals match with broad concepts in the ontology, while the detailed learning activities connect to specific relationships and topics within the educational structure.

The Backward Course Design Model and ontological reasoning work together effectively in engineering education, with the ontology-based reasoner built on established bottom-up and top-down approaches [32]. The Backward Course Design Model helps structure how courses are developed, while ontological reasoning helps organize engineering concepts and their connections. This combined approach creates a flexible system that can adapt to meet the changing needs of both students and instructors, leading to better educational outcomes at universities.

2.4 Stakeholder Perspectives

From a pedagogical perspective, students tend to focus on their personal goals and interests when managing their academic progress, while instructors tend to focus more on learning outcomes and program-level objectives. These distinct perspectives naturally inform our ontology design approach: the bottom-up student perspective helps to create practical relationships between courses, prerequisites, and skills development, whereas the top-down instructor perspective helps align course objectives and overall curriculum structure [32]. This dual perspective ensures the ontology-based reasoner can become comprehensive in its knowledge representation, making it valuable for all stakeholders in the education process.

One of the key advantages of the ontology-based framework is the flexibility it offers to students pursuing diverse academic and industrial projects based on their chosen career paths. Prior work on supervised homework sessions in aerospace structural mechanics courses demonstrated that structured, outside-the-classroom academic support can enhance student engagement and performance, particularly in complex topics such as stress and shear flow [33, 34]. Similarly, if students are interested in multidisciplinary and advanced research fields (e.g., heat shields for hypersonic vehicles) as academics, the students can use the ontology-based reasoner to identify specific prerequisite knowledge (e.g., materials used in extreme temperatures and ablation during hypersonic flight) and find where this knowledge is covered within the engineering disciplines (e.g., materials courses within aerospace engineering, mechanical engineering, civil engineering, and materials engineering).

For students preparing for careers in industry, the framework emphasizes practical applications of aerospace engineering principles. For example, the ontology-based reasoner can guide students through engineering design trade-offs, such as recommending Latin hypercube sampling over full factorial sampling for efficient experimental design. This helps students understand the balance between analysis fidelity and practical constraints while developing skills for real-world engineering challenges.

From the instructor's perspective, the ontology-based reasoner enables the design of modular courses that align with individual students' academic needs while maintaining curriculum coherence. This modular approach allows instructors to map relationships between course materials and prerequisite knowledge, bring in subject matter experts for specialized topics, and create customized content paths. By enabling this flexibility in course design and delivery, the framework helps instructors develop targeted curricula that enhance the educational experience while meeting both individual student needs and broader program objectives.

2.5 Research Gap and Proposed Solution

While decision-making tools are established in fields such as defense applications [35] and space applications [36, 37], and recent developments show increasing adoption of reasoner capabilities in both aerospace engineering practice [38] and university research [39], aerospace engineering education has yet to fully leverage these advancements. For example, structured decision frameworks developed for managing intellectual property in defense acquisition highlight how formal reasoning architectures support complex stakeholder negotiation and strategy formulation [40, 41, 42]. Drawing inspiration from these findings, and to address this gap in engineering education, we propose a novel dual-approach reasoner that combines Logical Consequences and Word Embeddings, designed specifically for academic decision-making. This tool integrates both bottom-up and top-down methodologies [32]. The bottom-up approach empowers students in their academic planning, while the top-down approach provides instructors with increased flexibility in curriculum development and advising.

3 Theoretical Framework on Ontology-based Reasoner

Logical consequences and word embeddings are distinct concepts. In logical consequences, a path starts from what we know to be true and follows a clear path from start to finish. Logical consequences describe the relationship between statements. In these statements, one statement follows the other statement. On the other hand, in word embeddings, words are represented numerically in a vector format. Word embeddings are an essential part of LLMs and help computers understand words by seeing how they connect to other words.

While these two methods work differently, they help us understand information. In the following sections, we will examine these ideas more closely. We will see how logical consequences help us think clearly (aids proper thinking) and how word embeddings help computers understand human language (enables computers to process human language).

3.1 Overall View of Data Propagation

The two distinct approaches of logical consequences and word embeddings do not compete. Fig. 3 describes two paths of ontology-based reasoner. They complement each other in system design since they serve different purposes. Logical consequences (Path 1) provide explicit reasoning capabilities through well-defined rules and relationships, enabling precise query processing and derivation in structured knowledge representations, meaning that the results are presented in a structured format rather than natural language. Word embeddings (Path 2) offer robust natural language understanding, allowing the system to process unstructured text and capture nuanced relationships. When integrated, these approaches combine the strengths of symbolic reasoning with probabilistic language processing capability. Logical consequences (Path 1) ensure reliable rule-based operations and structured outputs, while word embeddings (Path 2) facilitate flexible natural language understanding.

3.2 Path 1: Logical Consequences (SPARQL)

3.2.1 Introduction to Logical Consequences

Logical consequences form the basis of systematic reasoning. When we have a set of true statements, logical consequences are the conclusions that must follow from those statements. This straightforward principle helps us draw reliable conclusions from known facts, providing a foundation for clear and structured thinking.

Example 1: Consider the following reasoning:

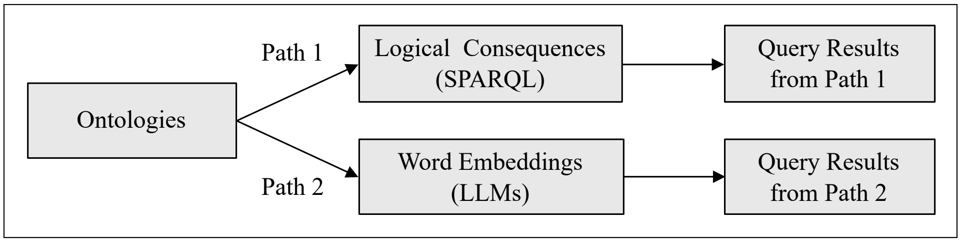

Figure 3: Ontology-based Reasoner Paths: Logical Consequences vs. Word Embeddings, Illustrating the Complementary Approaches to Educational Knowledge Processing

> **Premises:** All dogs are mammals, and Commander is a dog.
> **Reasoning:** From these premises, it follows logically that:
> **Conclusion:** Commander is a mammal.

In Example 1, the premises' truth guarantees the conclusion's truth. This is the essence of logical consequences: the conclusion is not merely likely or probable but necessarily true given the premises. That is, all B (dogs) are A (mammals); C (Commander) is B; therefore, C is A.

Example 2: Consider another example that is more in line with the theme of the current research:

> **Premises:** A student cannot enroll in required upper-division engineering courses without completing calculus prerequisites, and no student can receive an engineering degree without completing required upper-division engineering courses.
> **Reasoning:** From these premises, it follows logically that:
> **Conclusion:** No student can receive an engineering degree without completing calculus prerequisites.

In Example 2, the logical consequences demonstrate that if calculus prerequisites are required for upper-division engineering courses, and these engineering courses are required for an engineering degree, then by necessity, calculus prerequisites are required for an engineering degree. The reasoning chain works like a series of connected requirements - if calculus is needed for advanced courses and advanced courses are needed for the degree, then calculus must be needed for the degree. That is, if A (calculus) is required for B (upper-division courses), and B is required for C (engineering degree), then A must be required for C.

When comparing Examples 1 and 2, they follow a similar logical consequence structure. However, these examples show distinct types of relationships. Example 1 illustrates set membership, where all members of the set "dogs" are the members of the more extensive set "mammals." In contrast, Example 2 represents requirements in academic sequential progression, where the passing grades in calculus courses serve as a foundational requirement (prerequisite) for advanced engineering courses. However, despite their different domains, both examples represent the same principle of logical consequences: when A is connected to B, and B to C, then A must be connected to C. This logical framework maintains its validity, whether applied to set facts of animal classification or curricular prerequisites in university education. Thus, these examples demonstrate the versatility of logical consequences across different contexts.

The systematic, step-by-step approach of logical consequences mirrors the SoS-inspired frameworks that emphasized the DAI process to break down complex engineering problems into manageable components [28]. The logical consequences approach presented in our current work provides students with a structured pathway to navigate complex engineering problems, similar to how the DAI framework creates organized problem-solving strategies in aerospace structural mechanics education.

3.2.2 Key Characteristics of Logical Consequences

Logical consequences have some key features that make them a reliable way of reasoning [43]. First, they provide certainty: when we start with true statements, we can be completely confident in our conclusions. Second, they work the same way regardless of the topic: the reasoning patterns remain consistent whether we are talking about engineering, science, or any other field. Third, logical consequences rely on clear reasoning rather than physical evidence: we can reach valid conclusions just by carefully thinking through the relationships between statements [43].

The logical consequences use SPARQL (SPARQL Protocol and RDF Query Language), a W3C-standardized query language for accessing Resource Description Framework (RDF) data [44]. SPARQL is a recursive acronym where "SPARQL" is part of its own definition, and both SPARQL and RDF were developed by W3C [45, 46]. SPARQL enables precise querying of ontological relationships through a structured format, allowing users to extract specific information based on formalized logical pathways. While SPARQL requires users to understand its specific query syntax and structure, it provides robust and accurate results by following explicit logical pathways defined in the ontology [47].

3.2.3 Summary of Logical Consequences

Logical consequences are a fundamental concept in logic that describes when a conclusion must follow from a given premise with absolute certainty [43]. Unlike scientific observations that rely on real-world evidence, logical consequences work through the pure structure of arguments, regardless of their specific content [43]. For example, if we know all dogs are mammals and Commander is a dog, then Commander must be a mammal. This conclusion is guaranteed by the logical form alone, not by any real-world knowledge about dogs or mammals. This combination of necessity, structural validity, and independence from empirical evidence makes logical consequences a cornerstone of rigorous reasoning [43]. SPARQL implements these logical consequence principles in practical applications through a structured format, enabling precise querying of relationships based on formalized logical pathways. Understanding logical consequences is essential for implementing SPARQL queries in practical applications. While this approach offers precision and explicit rule-based reasoning, it can benefit from complementary capabilities to handle the natural language and semantic richness found in educational contexts.

3.3 Path 2: Word Embeddings (LLMs)

3.3.1 Introduction to Word Embeddings

Word embeddings are a key component of LLMs, serving as the foundation for converting textual data into numerical representations that machines can process. Fig. 4 depicts the fundamental idea behind word embeddings, where words are mapped to high-dimensional vectors in a continuous space, enabling mathematical operations on textual data. For instance, in a well-constructed embedding space, technical terms from related engineering domains would be positioned closer together (e.g., clustering terms like “stress,” “strain,” “deformation,” and “mechanics” in aerospace solid mechanics) in proximity to reflect their interconnected conceptual relationships. This transformation allows LLMs to capture semantic relationships between technical vocabulary and process language in a computationally efficient manner.

Example 3: Consider an example in the context of engineering prerequisites:

> **Vector Space:** Terms like “calculus,” “differential equations,” and “numerical methods” are embedded near each other.
> **Relationship:** These mathematical concepts build upon each other in engineering education.
> **Result:** The LLMs can recognize the progressive nature of these topics and their interdependencies.

Example 4: Consider another example in the context of engineering concepts:

> **Vector Space:** The terms “aerodynamics,” “fluid dynamics,” and “compressible flow” are mapped close together in the vector space.
> **Relationship:** These engineering terms share similar contexts in aerospace engineering education.
> **Result:** The LLMs can understand these terms as being closely related and likely appearing in similar educational contexts

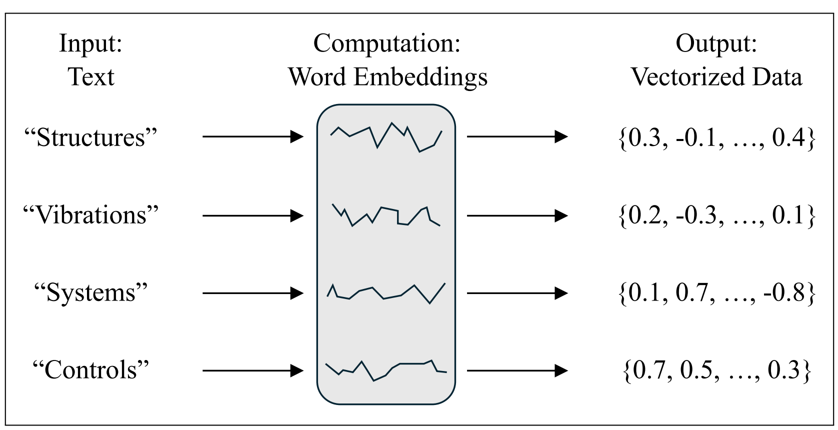

Figure 4: Word Embeddings

3.3.2 Key Characteristics of Word Embeddings

Word embeddings have several key features that make them especially useful for LLMs in engineering education. They help computers understand relationships between words in ways that mirror how we think about them. When words have similar meanings, like different types of engineering topics such as structures, vibrations, systems, and controls, word embeddings place them close together in their computational space. They can also understand relationships between pairs of related concepts, such as how calculus relates to differentiation and how algebra relates to equation solving.

Word embeddings are also smart about understanding context. They can figure out that the same word might mean different things in different situations (e.g., "stress" in mechanics versus "stress testing" in systems engineering). This ability to grasp different meanings based on context is particularly valuable in engineering education, where technical terms often have specific meanings depending on their field.

3.3.3 Summary of Word Embeddings

Word embeddings convert text into numbers (i.e., converting texts into vectors) while preserving the meaning relationships between words. In LLMs, they serve as a bridge between human language and computer calculations, enabling machines to understand and process complex engineering concepts. Unlike logical consequences, which follow strict rules, word embeddings capture the subtle nuances of language and context in engineering education. This number-based representation has transformed how computers process language by capturing not just the literal meaning of words but also their relationships, contexts, and domain-specific applications in engineering education. When combined with the structured precision of logical consequences, word embeddings can create a comprehensive approach that balances rigorous reasoning with flexible understanding.

4 Methodology

4.1 Ontology

The implementation of the ontology-based reasoner presents a novel approach to structuring and optimizing aerospace engineering curricula by leveraging relationships between fundamental concepts, courses, and classes. This ontology-based reasoner helps students develop personalized study plans based on their academic interests and assists academic advisors when reviewing proposed curricula. The reasoner follows a logical consequence framework built upon the hierarchical nature of engineering knowledge acquisition, where courses typically progress from fundamental concepts (first- and second-year engineering) to advanced applications (third- and fourth-year engineering).

This methodology is structured upon two fundamental layers: concepts (Fig. 5) and courses (Fig. 6), each forming a crucial component in the educational architecture. Together, these layers create a comprehensive framework that maps the progression of knowledge throughout the aerospace engineering curriculum.

4.1.1 Concepts in Aerospace Engineering

Fig. 5 depicts concepts. The framework begins with the basic concepts in aerospace engineering and the fundamental mathematical concepts such as eigenvalue decomposition and differential equations. These basic concepts can then easily be transferred to more advanced topics such as optimization, entropy, and electromagnetism. The relationships between these concepts also become apparent in the application of finite element analysis and control theory, where several principles are required to form a solution to a problem.

```
[Concepts]
  ├── [Eigenvalue Decomposition]
  ├── [Optimization]
  ├── [Entropy]
  ├── [Electromagnetism]
  ├── [Linear Programming]
  ├── [Finite Element Analysis (FEA)]
  ├── [Control Theory]
  ├── [Fluid Dynamics]
  ├── [Kinematics]
  ├── [Vibration Analysis]
  ├── [Linear Algebra]
  ├── [Differential Equations]
```

Figure 5: Concepts in Aerospace Engineering

4.1.2 Courses in Aerospace Engineering

Fig. 6 shows courses. The next layer contains specific subjects that use these fundamental concepts in learning activities. For instance, engineering mathematics provides the necessary tools for advanced topics. From there, the students move on to application-based courses in areas such as thermodynamics, mechanics of materials, and fluid mechanics, which link theoretical ideas to aerospace. This proper sequencing of topics helps the students build on the knowledge they have already gained before moving to the next level of subject matter .

Thus, the ontology-based reasoner can help LLMs identify gaps in the knowledge, ensure understanding at the appropriate level, and provide focused responses at every level of learning. This is because they are able to identify the relationships between aerospace engineering concepts in a way that is beneficial to the development of education in this subject area.

4.2 Concept-Course Integration in Aerospace Engineering

Fig. 7 depicts the combined and interconnected nature of concepts and courses in aerospace engineering education. The diagram maps how fundamental mathematical and engineering concepts flow into specialized aerospace courses (shown in green). This visualization demonstrates how basic principles, such as eigenvalue decomposition and differential equations, serve as building blocks for advanced courses like thermodynamics and fluid mechanics. The visualization depicts “teaches” and “uses” relationships between nodes, revealing the prerequisites and knowledge transfer pathways, and highlighting how theoretical foundations support application-based learning throughout the aerospace engineering curriculum.

```
[Courses]
 ├── [Engineering Mathematics]
 │   ├── [Linear Algebra] --Teaches--> [Eigenvalue Decomposition]
 │   ├── [Differential Equations] --Teaches--> [Optimization]
 │   ├── [Optimization Methods] --Teaches--> [Linear Programming]
 ├── [Thermodynamics]
 │   ├── [Laws of Thermodynamics] --Teaches--> [Entropy]
 │   ├── [Phase Change] --Teaches--> [Entropy]
 ├── [Mechanics of Materials]
 │   ├── [Stress and Strain] --Uses--> [Linear Algebra]
 │   ├── [Elasticity and Plasticity] --Uses--> [Finite Element Analysis (FEA)]
 ├── [Control Systems]
 │   ├── [Linear Systems] --Teaches--> [Linear Algebra]
 │   ├── [Feedback and Stability] --Uses--> [Eigenvalue Decomposition]
 │   ├── [Optimization in Control] --Uses--> [Optimization]
 ├── [Fluid Mechanics]
 │   ├── [Fluid Properties] --Teaches--> [Fluid Dynamics]
 │   ├── [Navier-Stokes Equations] --Teaches--> [Fluid Dynamics]
 │   ├── [Optimization in Fluid Flow] --Uses--> [Optimization]
 ├── [Electromagnetism]
 │   ├── [Electromagnetic Fields] --Teaches--> [Electromagnetism]
 │   ├── [Maxwell's Equations] --Teaches--> [Electromagnetism]
 ├── [Vibration Analysis]
 │   ├── [Mechanical Vibrations] --Uses--> [Eigenvalue Decomposition]
 │   ├── [Modal Analysis] --Uses--> [Eigenvalue Decomposition]
 │   ├── [Control of Vibrations] --Uses--> [Optimization]
```

Figure 6: Courses in Aerospace Engineering

```
Classes:
├── Concept
│   ├── Eigenvalue Decomposition (C1)
│   ├── Differential Equation (C2)
│   ├── Linear Algebra (C3)
│   ├── Modal Analysis (C4)
│   └── System Stability (C5)
└── Course
    ├── Engineering Mathematics I: Linear Algebra (CO1)
    │   ├── teaches: C1, C3
    │   └── uses:
    ├── Vibration Analysis (CO2)
    │   ├── teaches: C4
    │   └── uses: C1
    ├── Control Theory (CO3)
    │   ├── teaches: C5
    │   └── uses: C1, C2
    ├── Engineering Mathematics II: Differential Equations (CO4)
        ├── teaches: C2
        └── uses:
```

Figure 7: Combined Concepts and Courses in Aerospace Engineering

4.3 Implementation Approaches

The ontological framework described above can be implemented through two distinct yet complementary approaches. The first approach utilizes SPARQL queries to establish logical consequences within the knowledge structure, enabling precise retrieval of relationships between concepts and courses. The second approach leverages Large Language Models (LLMs) through word embeddings to capture semantic relationships in the educational framework. These two paths offer different strengths in implementing the ontology: SPARQL provides explicit, rule-based querying capabilities, whereas LLMs can identify implicit connections and semantic similarities within the educational content. Together, they provide a robust foundation for navigating and utilizing the aerospace engineering curriculum ontology.

4.4 Path 1: Logical Consequence (SPARQL)

To demonstrate the framework for logical consequences (SPARQL), we queried courses that teach and use eigenvalue decomposition in our Ontology for Engineering Education (OEE), as shown in Fig. 8. The query showcases not only the relevant courses but also additional ontological information. For instance, the current version displays semester offerings. However, the ontological structure can be further expanded to include additional metadata. From a bottom-up student perspective, this could include prerequisites, credit hours, required materials, and program requirements. From a top-down instructor/administrator perspective, the metadata could encompass course capacity, teaching assignments, equipment requirements (especially for hands-on labs), and program accreditation details. This example illustrates how the ontological structure enables retrieval of both direct relationships (courses teaching a specific concept) and related metadata from multiple stakeholder perspectives, demonstrating the powerful querying capabilities of the logical consequence approach.

```
PREFIX : <http://www.example.org/curriculum#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?course ?courseName ?semester WHERE {
  ?course a OEE:Course .
  ?course OEE:teaches ?concept .
  ?concept rdfs:label "Eigenvalue Decomposition" .
  ?course rdfs:label ?courseName .
  ?course :offerSemester ?semester .
}
```

Figure 8: Example of SPARQL Query

4.5 Path 2: Word Embeddings (LLMs)

To demonstrate the framework for word embeddings (LLMs), we queried courses that teach and use eigenvalue decomposition in our ontologies, as shown in Fig. 9. This input file represents educational ontologies in aerospace engineering, defining relationships between fundamental concepts (C1-C5) and courses (CO1-CO4). It uses a structured format where each line describes either a concept definition (e.g., "Eigenvalue Decomposition"), a course definition (e.g., "Engineering Mathematics I"), or relationships between them using "teaches" and "uses" predicates. This input file shows how courses build upon and teach different concepts - for instance, Engineering Mathematics I teaches eigenvalue decomposition and linear algebra, while Control Theory teaches system stability while utilizing concepts from eigenvalue decomposition and differential equations. This structure helps map the prerequisite relationships and knowledge flow through the curriculum.

```
C1 is a concept
Eigenvalue Decomposition = C1
C2 is a concept
Differential Equation = C2
C3 is a concept
Linear Algebra = C3
C4 is a concept
Modal Analysis = C4
C5 is a concept
System Stability = C5

CO1 is a course
Engineering Mathematics I: Linear Algebra = CO1
CO1 teaches C1, C3
CO1 uses nothing

CO2 is a course
Vibration Analysis = CO2
CO2 teaches C4
CO2 uses C1

CO3 is a course
Control Theory = CO3
CO3 teaches C5
CO3 uses C1, C2

CO4 is a course
Engineering Mathematics II: Differential Equations = CO4
CO4 teaches C2
CO4 uses nothing
```

Figure 9: Example of LLM Query

5 Results

5.1 Path 1: Logical Consequences (SPARQL)

Fig. 10 depicts results based on the SPARQL query shown in Fig. 8. Fig. 10a indicates courses in which teaching eigenvalue decomposition is a core concept. Fig. 10b demonstrates courses that apply eigenvalue decomposition in their curriculum.

5.2 Path 2: Word Embeddings (LLMs)

Fig. 11 depicts results based on the LLM query shown in Fig. 9. Fig. 11a indicates courses in which teaching eigenvalue decomposition is a core concept. Fig. 11b demonstrates courses that use eigenvalue decomposition in their curriculum.

?CourseSelected	?ConceptName	?offerSemester
Engineering Mathematics	Eigenvalue decomposition	Fall

(a) Courses Teaching Eigenvalue Decomposition as a Core Concept

?CourseSelected	?ConceptName	?offerSemester
Control System	Eigenvalue decomposition	Spring
Vibration Analysis	Eigenvalue decomposition	Fall

(b) Courses Applying Eigenvalue Decomposition in Their Curriculum

Figure 10: Results based on SPARQL Query Depicted in Fig. 8

Question:
What courses teach eigenvalue decomposition?

Answer:
The course that teaches Eigenvalue Decomposition (C1) is:
Engineering Mathematics I: Linear Algebra (CO1)

(a) Courses Teaching Eigenvalue Decomposition as a Core Concept

Question:
What courses apply eigenvalue decomposition?

Answer:
The courses that apply Eigenvalue Decomposition (C1) are:
Vibration Analysis (CO2)
Control Theory (CO3)

(b) Courses Applying Eigenvalue Decomposition in Their Curriculum

Figure 11: Results based on LLM Query Depicted in Fig. 9

6 Discussion

6.1 Comparative Analysis: Logical Consequences vs. Word Embeddings

When comparing logical consequences (SPARQL) with word embeddings (LLMs), we observed fundamental differences in both their underlying approaches and their practical results in curriculum analysis. The SPARQL-based approach operates on precise, structured data relationships defined in the ontology, using explicit rules and formal logic to follow deterministic query paths. This is evident in our results (Fig. 10), where the SPARQL query produced clearly structured outputs differentiating between courses that teach eigenvalue decomposition as a core concept (Fig. 10a) and those that apply it (Fig. 10b). This explicit differentiation proves particularly valuable for curriculum planning and prerequisite mapping.

In contrast, LLMs use probabilistic relationships between words in vector space, where relationships are learned from training data rather than explicitly defined. The LLM-based results (Fig. 11) demonstrate this different approach, presenting similar course classifications but through natural language processing. While both approaches identified Engineering Mathematics I as a primary course for teaching eigenvalue decomposition, the LLM approach provided additional contextual insights about concept application in courses like Control Theory.

These differences reflect fundamental trade-offs in each approach. SPARQL delivers results in a structured, precise manner but requires users to understand specific query syntax. To illustrate this difference: if an ontology incorrectly states, “All birds can fly,” SPARQL will logically but incorrectly deduce that ostriches and kiwis can fly, adhering strictly to the defined rule. On the other hand, LLMs use probability-based pattern matching and offer a more user-friendly interface through natural language processing. When asked about birds and flying, an LLM might correctly identify that ostriches and kiwis do not fly despite being birds, based on learned patterns. However, LLMs face challenges with reliability, particularly with large datasets where their performance can vary for uncommon cases and occasionally lead to incorrect generalizations. As datasets grow, the vector space becomes increasingly crowded, leading to more ambiguous semantic relationships.

These contrasting characteristics suggest complementary strengths. While SPARQL excels in precise relationship mapping and maintaining consistency regardless of dataset size, LLMs can identify subtle conceptual connections that might not be explicitly coded in the ontology. This complementarity suggests that an integrated approach using both paths could provide the most comprehensive tool for curriculum development and academic advising. The SPARQL path ensures rigorous logic-based course relationships, while the LLM path enriches this structure with semantic context and potential relationships that traditional prerequisite structures might overlook. These complementary strengths of both approaches become particularly evident when examining their applications from different user perspectives, as discussed in the following sections.

6.2 Bottom-up Student Perspective

The ontology-based reasoner in this bottom-up approach leverages both logical consequences and word embeddings to map students’ interests to relevant knowledge areas. Through SPARQL queries, the reasoner can traverse the ontological relationships to identify both direct and indirect connections between concepts, courses, and applications. Complementing this, the LLM component enhances the system's ability to understand and process natural language expressions of student interests and goals, making the interaction more intuitive. This dual approach enables the reasoner to provide comprehensive pathways that connect students' career interests with the necessary educational components while maintaining accessibility and precision. This implementation demonstrates how the trade-offs identified in our comparative analysis manifest in practical student applications.

The systematic approach of our ontology-based reasoner mirrors the DAI process [28] as students define their learning goals, abstract necessary knowledge components, and implement their understanding through practical applications. Our results showed that the LLM successfully developed individualized learning pathways by aligning concepts, courses, and classes with student needs. The system demonstrated particular effectiveness in identifying and addressing knowledge gaps. For example, when students showed weakness in linear algebra fundamentals, the system automatically prescribed targeted supplementary materials before introducing applications in aerospace structures or flight dynamics. This personalization is particularly valuable for students who have clear career goals and need to understand the specific prerequisite knowledge required for their chosen path.

Our implementation demonstrated effective integration of concepts and courses from a student perspective. The system successfully identified natural connections between foundational concepts and their applications, helping students understand why certain prerequisites are necessary. For example, eigenvalue decomposition's crucial role in vibration analysis and control theory was automatically linked with engineering mathematics courses, providing students with a clear motivation for learning these fundamental concepts.

From the student perspective, the framework's adaptability to curriculum changes ensures they stay current with industry developments. As new technologies and concepts emerge in fields such as materials science and robotics, students can see how their coursework connects to real-world applications, maintaining the relevance of their education to industry needs. This student-centered perspective illustrates how combining the structured precision of SPARQL with the flexible understanding of LLMs creates an effective educational support system, leading us to examine how these same approaches serve instructor needs.

6.3 Top-down Instructor Perspective

The ontology-based reasoner supports this top-down approach through complementary mechanisms of logical consequences and natural language processing. Through SPARQL queries, instructors can systematically identify knowledge gaps, track prerequisite chains, and ensure comprehensive coverage of fundamental concepts across multiple courses. The LLM component enriches this capability by processing instructors' natural language queries about curriculum relationships and offering context-aware suggestions for course content optimization. For example, an instructor teaching advanced aerodynamics can use SPARQL queries to verify that all necessary mathematical foundations, such as partial differential equations and computational methods, are properly covered in prerequisite courses while using the LLM to explore potential connections to emerging aerospace technologies and identify subtle concept relationships that might not be explicitly coded in the ontology. This combination of precise semantic querying and flexible natural language processing helps instructors maintain curriculum coherence while enabling adaptive responses to evolving educational needs. This implementation from the instructor's perspective further validates the complementary nature of our dual-approach system discussed in the comparative analysis.

Instructors can leverage the DAI framework [28] to systematically define learning objectives, abstract core concepts, and guide implementation while using the ontology-based reasoner to validate curriculum coherence. The framework proved particularly effective in supporting instructors' curriculum optimization efforts. By analyzing prerequisite relationships, the system accurately identified critical learning sequences, such as the necessity of mastering differential equations before progressing to fluid dynamics or control systems. This enables instructors to make data-driven decisions about course sequencing and content coverage.

From the instructor's perspective, the framework's ability to adapt to curriculum evolution is crucial for maintaining educational quality. The system allows instructors to integrate cutting-edge knowledge and technologies while preserving logical consistency in the curriculum. This enables them to effectively connect classroom learning with industry developments, ensuring their teaching remains relevant and comprehensive. These instructor experiences with both SPARQL and LLM approaches demonstrate how their respective strengths can be leveraged effectively in curriculum development, providing a practical validation of our earlier comparative analysis.

6.4 Summary of Discussion

This discussion analyzes the implementation of an educational ontology system in aerospace engineering through the following main aspects. First, it compares two technical approaches: logical consequences (SPARQL) queries, which offer precise but rigid rule-based results, and word embeddings (LLMs), which provide more flexible natural language processing but with potential reliability challenges. Second, from a bottom-up student perspective, the system successfully creates personalized learning pathways and helps students understand prerequisite relationships, which is particularly beneficial for those with clear career paths. Finally, from a top-down instructor perspective, the system enables effective curriculum optimization and maintenance, allowing instructors to verify prerequisite coverage and adapt to emerging technologies while maintaining educational coherence. Throughout these aspects, the discussion demonstrates how the approaches of logical consequence and word embeddings can contribute to creating a comprehensive educational framework that serves both students' and instructors' needs in aerospace engineering education.

7 Conclusions and Future Work

This paper has demonstrated how an ontology-based reasoner can leverage two distinct paths, logical consequences (SPARQL) and word embeddings (LLMs), to create robust approaches for aerospace engineering education. This integration extends the DAI process [28] into a comprehensive educational framework that supports both structured reasoning and flexible understanding in aerospace engineering education. The evaluation of SPARQL's structured, rule-based queries alongside LLMs' natural language processing capabilities showed how each path effectively addresses distinct educational needs, enabling personalized learning pathways and clear prerequisite understanding for students while providing curriculum optimization tools and technology adaptation capabilities for instructors. Through our implementation and use cases, we have shown that these complementary approaches support both precise, logical reasoning and flexible natural language understanding, advancing aerospace engineering education by creating a more comprehensive and adaptable learning environment.

The significance of this work extends beyond its specific implementation. By bridging formal reasoning with natural language capabilities, our approach addresses a fundamental challenge in engineering education: balancing structured knowledge pathways with contextual understanding. This ontology-based reasoner represents a step toward educational systems that adapt to aerospace engineering's interdisciplinary nature while maintaining technical rigor. Such frameworks will become increasingly vital as curricula evolve to prepare students for complex, real-world challenges.

Future research should explore integrating this framework with Model-Based Systems Engineering (MBSE) [39, 48]. This integration could extend our semantic querying capabilities from educational contexts to practical aerospace systems development, creating a bridge between theoretical concepts and their implementation in real-world projects. The resulting framework would serve as both an educational tool and a systems engineering platform, contributing to more effective teaching of aerospace systems engineering while facilitating the transition from academic learning to professional practice.

References

[1] D. A. DeLaurentis, K. Moolchandani, and C. Guariniello, *System of systems modeling and analysis*. CRC Press, 2022.

[2] B. R. Schatz and J. B. Hardin, "Ncsa mosaic and the world wide web: global hypermedia protocols for the internet," *Science*, vol. 265, no. 5174, pp. 895–901, 1994.

[3] T. Berners-Lee, "Www: Past, present, and future," *Computer*, vol. 29, no. 10, pp. 69–77, 1996.

[4] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of computer-mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[5] C. Wagner and A. Majchrzak, "Enabling customer-centricity using wikis and the wiki way," *Journal of management information systems*, vol. 23, no. 3, pp. 17–43, 2006.

[6] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *2008 16th Interntional Workshop on Quality of Service*. IEEE, 2008, pp. 229–238.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[10] R. A. Diogo, N. dos Santos, and E. F. Loures, "Digital transformation of engineering education for smart education: a systematic literature review," *Reliability Modeling in Industry 4.0*, pp. 407–438, 2023.

[11] E. F. Crawley, D. R. Brodeur, and D. H. Soderholm, "The education of future aeronautical engineers: Conceiving, designing, implementing and operating," *Journal of Science Education and Technology*, vol. 17, pp. 138–151, 2008.

[12] A. Zubiaga, "Natural language processing in the era of large language models," p. 1350306, 2024.

[13] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[14] N. Guarino, *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. IOS press, 1998, vol. 46.

[15] M. Gómez-Torrente, "Tarski on logical consequence," *Notre Dame Journal of Formal Logic*, vol. 37, no. 1, pp. 125–151, 1996.

[16] G. Ray, "Logical consequence: A defense of tarski," *Journal of Philosophical Logic*, vol. 25, pp. 617–677, 1996.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[19] F. Neuhaus, "Ontologies in the era of large language models–a perspective," *Applied ontology*, vol. 18, no. 4, pp. 399–407, 2023.

[20] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review*, vol. 11, no. 2, pp. 93–136, 1996.

[21] N. F. Noy, D. L. McGuinness *et al.*, "Ontology development 101: A guide to creating your first ontology," 2001.

[22] D. DeLaurentis and R. K. Callaway, "A system-of-systems perspective for public policy decisions," *Review of Policy research*, vol. 21, no. 6, pp. 829–837, 2004.

[23] L. Knöös Franzén, I. Staack, C. Jouannet, and P. Krus, "An ontological approach to system of systems engineering in product development," in *proceedings of the 10th Aerospace Technology Congress*, vol. 162. Linköping University Electronic Press, 2019, pp. 35–44.

[24] I. Sanya and E. Shehab, "An ontology framework for developing platform-independent knowledge-based engineering systems in the aerospace industry," *International Journal of Production Research*, vol. 52, no. 20, pp. 6192–6215, 2014.

[25] M. Ast, M. Glas, T. Roehm, and V. Luftfahrt, *Creating an ontology for aircraft design*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2014.

[26] W. Tsutsui, K. Park, C. Sculley, I. Copenhaver, M. Gonzalez, W. Chen, G. Takahashi, and M. Sangid, "Enhancing students' understanding of deformation and stress in aerospace structures education via virtual labs," in *2022 ASEE Annual Conference & Exposition*, 2022.

[27] W. Tsutsui, E. J. Williamson, K. Park, and M. D. Sangid, "Pedagogy improvement in aerospace structures education using virtual labs: Before, during, and after the covid-19 school closures and remote learning," in *2021 ASEE Virtual Annual Conference Content Access*, 2021.

[28] W. Tsutsui and D. DeLaurentis, "A system-of-systems inspired framework to enhance aerospace structural mechanics education," in *2023 ASEE Annual Conference & Exposition*, 2023.

[29] G. P. Wiggins and J. McTighe, *Understanding by design*. Association for Supervision and Curriculum Development, 1998.

[30] D. L. Carlson and P. A. Marshall, "Learning the science of research, learning the art of teaching: Planning backwards in a college genetics course," *Bioscience Education*, vol. 13, no. 1, pp. 1–9, 2009.

[31] W. Tsutsui, R. D. Lopez-Parra, G. S. Coutinho, A. W. Mello, M. D. Sangid, and T. J. Moore, "The implementation of virtual labs in aerospace structures education," in *2020 ASEE Virtual Annual Conference Content Access*, 2020.

[32] W. Tsutsui, V. Zeltsman, T. S. Adams, J. H. Panchal, and D. DeLaurentis, "Leveraging ontologies in engineering education: Top-down and bottom-up approaches," in *2024 ASEE Annual Conference & Exposition*, 2024.

[33] W. Tsutsui and M. C. Loui, "The effectiveness of weekly supervised homework sessions in an aerospace structural mechanics course," in *2016 ASEE Annual Conference & Exposition*, 2016.

[34] W. Tsutsui and M. Loui, "The impact of supervised homework sessions and sat-math scores on academic performance in an advanced undergraduate course," in *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2016, pp. 1–8.

[35] W. Tsutsui, Q. Shi, I. Walter, A. Wei, C. Williams, D. DeLaurentis, and J. Panchal, "Decision making for additive manufacturing in sustainable defense acquisition," *Naval Engineers Journal*, vol. 135, no. 4, pp. 47–58, 2023.

[36] Q. Shi, W. Tsutsui, D. Bekdache, J. H. Panchal, and D. DeLaurentis, "A system-of-systems (sos) perspective on additive manufacturing decisions for space applications," in *2022 17th Annual System of Systems Engineering Conference (SOSE)*. IEEE, 2022, pp. 282–288.

[37] Q. Shi, W. Tsutsui, I. Walter, J. Panchal, and D. DeLaurentis, "A decision support framework for additive manufacturing of space satellite systems," in *2023 IEEE Aerospace Conference*. IEEE, 2023, pp. 1–15.

[38] K. Amadori, C. Jouannet, and L. Knöös Franzén, "Ontology-based generation of onboard system architectures," in *AIAA SCITECH 2024 Forum*, 2024, p. 1088.

[39] S. Sinha Roy, Y. Huang, D. Bekdache, T.-Y. Fung, B. Beck, C. Guariniello, and D. DeLaurentis, “From ontology to system architecture: An mbse approach toward the realization of urban air mobility,” in *34th Congress of the International Council of the Aeronautical Sciences (ICAS)*, 2024.

[40] W. Tsutsui, Q. Shi, D. Bekdache, P. Balasubramani, J. Panchal, S. Biller, and D. DeLaurentis, “Intellectual property strategies for additive manufacturing in defense acquisitions,” in *27th Annual Systems and Mission Engineering Conference, Norfolk, VA*, 2024.

[41] D. DeLaurentis, S. Biller, J. H. Panchal, Q. Shi, D. Bekdache, P. Balasubramani, and W. Tsutsui, “Pilot program design to test innovative approaches in negotiating intellectual property,” Acquisition Innovation Research Center (AIRC), WRT-1081.18, technical report AIRC-2024-TR-015, 2024.

[42] W. Tsutsui, Q. Shi, D. Bekdache, P. Balasubramani, J. H. Panchal, S. Biller, and D. DeLaurentis, “Balancing access and protection: A decision framework for additive manufacturing intellectual property rights in defense acquisition,” *Acquisition Research Program*, 2025.

[43] J. Etchemendy, “Tarski on truth and logical consequence,” *The Journal of Symbolic Logic*, vol. 53, no. 1, pp. 51–79, 1988.

[44] E. Prudhommeaux, “Sparql query language for rdf,” *http://www. w3. org/TR/rdf-sparql-query/*, 2008.

[45] W. W. W. Consortium *et al.*, “Sparql 1.1 overview,” *https://www.w3.org/TR/sparql11-overview/*, 2013.

[46] G. Klyne, “Resource description framework (rdf): Concepts and abstract syntax,” *http://www.w3.org/TR/rdf-concepts/*, 2004.

[47] J. Pérez, M. Arenas, and C. Gutierrez, “Semantics and complexity of sparql,” *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 3, pp. 1–45, 2009.

[48] Y. Huang, T.-Y. Fung, and D. A. DeLaurentis, “Addressing complexity in system of systems with graphrag: An ai-driven framework for dynamic data integration,” 2025, in review.