34th Annual INCOSE international symposium
hybrid event
Dublin, Ireland
July 2 - 6, 2024

TOWARDS A SYSTEMS ENGINEERING ONTOLOGY STACK

Joe Gregory
Department of Systems and Industrial Engineering
University of Arizona
joegregory@arizona.edu

Alejandro Salado
Department of Systems and Industrial Engineering
University of Arizona
alejandrosalado@arizona.edu

Abstract. Semantic Web Technologies (SWTs) provide an approach to the structuring and understanding of data. SWTs utilize ontologies, reasoners, and query languages to structure existing knowledge, validate knowledge, and infer new knowledge. Ontologies in particular play a central role in enabling reusability and interoperability between domains. A common way to organize ontologies and their dependencies is in a layered ontology stack. These layers often incorporate top-level, core and domain ontologies. Libraries of standard instances can also be used. Federating the conceptualization of a domain across upper- and lower-level ontologies improves the reusability of higher-level terminology in other domains, and therefore improves interoperability between them. The University of Arizona Ontology Stack (UAOS) is a layered, modular ontology stack that has been developed to support digital engineering activities at the University of Arizona. It is based on the Basic Formal Ontology (BFO), and currently comprises five core ontologies and 12 domain ontologies. The UAOS reuses existing ontologies and standards wherever possible. The core ontologies, for example, are based on the Common Core Ontologies, developed at CUBRC, and the Provenance Notation (PROV-N), a W3C standard. Domain ontologies include the System Architecture Ontology, based on ISO 42010, and the Orbits and Trajectories Ontology, based on CUBRC's Space Object Ontology. In this paper, we report on the development of the UAOS, present examples of how it has been used to support digital engineering research, discuss the challenges of integrating ontologies from multiple sources into a cohesive stack, and highlight topics of interest for future research.

Keywords. Semantic web technologies, ontology, digital engineering, verification, education.

Introduction

One area of digital engineering that has received significant attention in recent years is the digital thread (Zimmerman et al., 2019). The digital thread refers to the integration of digital information across the entire lifecycle of a system. This necessarily means that different engineering tools, each supporting some aspect of the system lifecycle, must be integrated in some way as to enable the seamless transfer and utilization of data. Today's industry leaders consider digital thread initiatives to be a top priority (Dertien & Hastings, 2021). Examples of this increasingly important digital integration effort include work on the traceability from requirements to models to simulation results (Buffoni et al., 2017; Mengist et al., 2021; Otter et al., 2015) and the integration of requirements and verification strategies (Gregory & Salado, 2023a, 2024b; Salado & Kannan, 2019). To support this, data integration and management technologies are crucial. Relational databases and graph databases are both useful approaches to representing data. However, if a user wishes to query highly connected data, graph databases have the potential to provide results more efficiently (Medhi & Baruah, 2017). Graph databases also provide greater flexibility, as new relationships can be added to the data without the need to restructure the schema each time (Batra & Tyagi, 2012). Graph-based data integration may be considered along two axes: technical interoperability and data interoperability (Maló, 2013). Technical interoperability regards the integration of traditionally disparate engineering data from multiple tools into a central integrated tool. Data interoperability regards the structure of the data once it has been integrated.

Semantic web technologies (SWTs) offer a powerful approach to data integration. They have the potential to capture the knowledge and domain-specific concepts necessary for understanding and analyzing complex systems. SWTs utilize ontologies, reasoners and query languages to structure existing knowledge, validate knowledge, and infer new knowledge (Gregory et al., 2021). Ontologies in particular enable the combination of deep domain knowledge and raw data to bridge across domains (Patel & Jain, 2021). SWTs also have the potential to add value to integrated datasets by ensuring that all system-related information is stored and configuration-managed in a central repository (Madni & Sievers, 2018). Datasets can be captured using Resource Description Framework (RDF) triple-stores.

In this paper, the authors present an ontology-based approach to data integration that is being used to support digital engineering at the University of Arizona (UA). The University of Arizona Ontology Stack (UAOS) is a layered, modular ontology stack spanning multiple engineering domains, available at (Gregory & Salado, 2023b). We describe the methodology that has been adopted to support this development, present the current status of the UAOS, investigate the application of the UAOS to three notional projects, and discuss possibilities for future work.

Methodology

The research and development of the UAOS was based on the design science research approach. The methodology is presented in Figure 1. Design science research is an approach that focuses on the development of an artifact to address a particular problem (Baskerville et al., 2018; Dresch et al., 2015; Nabukenya, 2012). The artifact is then evaluated in terms of its applicability to the problem that was initially identified (Pries-Heje et al., 2008). In 2007, Hevner published a "three cycle view of design science research" (A. R. Hevner, 2007). The 'rigor cycle' ensures that the artifact design is founded on robust mathematical and scientific theory and existing standards. The 'relevance cycle' ensures that the artifact remains relevant to the stakeholders through requirements, testing, and feedback. The 'design cycle' represents the process of designing and building the artifact while providing data to, and receiving feedback from, the other two cycles. The research was divided into five main tasks, described in this section.

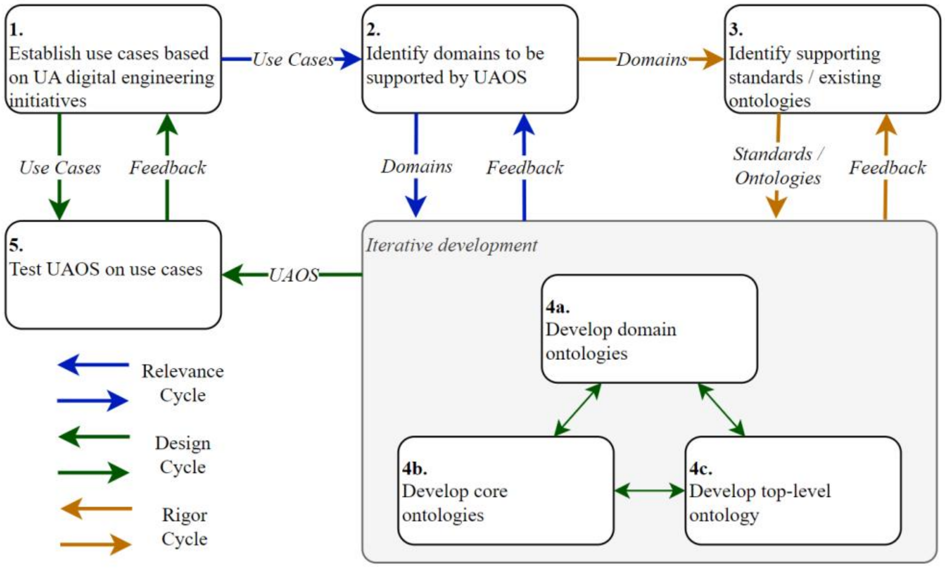

Fig. 1. Design science research methodology applied to UAOS development

Task 1: We established use cases based on digital engineering initiatives underway at UA. These use cases bound the scope of the development of the UAOS and provided a basis on which test cases could be built during development.

Task 2: We identified the domains that the UAOS must describe in order to support the use cases. Within each domain, we also identified the key terms and relations that were required to specifically cater to each use case. The goal at this stage has not been to develop comprehensive ontological representations of the domains of interest, but to precisely capture those aspects of the domain that are immediately required. As we continue development of the UAOS, we continue to refine and add to existing ontologies as well as produce new ones.

Task 3: Wherever possible, the ontologies that we developed to support the domains reused existing, tested ontologies or were based on relevant standards. The number of ontologies available for reuse depended on the top-level ontology that we chose, and so this influenced the decision that was made in Task 4.

Task 4: A common way to organize ontologies and their dependencies is in a layered ontology stack (Dunbar et al., 2023; Rudnicki et al., 2016). These layers often incorporate top-level, core and domain ontologies (Olsina, 2021). The UAOS followed this structure. This task included the selection of a suitable top-level ontology and the identification and development of any relevant core ontologies that could support the domain ontologies. We followed Noy's well-known ontology development methodology (Noy et al., 2001) to develop each domain ontology. This methodology states that, after initially considering the reuse of an existing ontology, important domain terms should be enumerated, classified in a taxonomy, and then relations should be added between the classes.

Task 5: A key part of design science research is the demonstration and evaluation of the product under design when applied to the original problem (A. Hevner & Chatterjee, 2010; A. R. Hevner et al., 2008). Throughout development, we applied the UAOS to notional datasets that represent one or more use cases and used the results to inform further development of the UAOS.

Digital Engineering at the University of Arizona

The primary purpose of the UAOS is to support digital engineering initiatives at UA. Accordingly, the objective of *Task 1* was to identify use cases within those initiatives. There are two main research activities currently under investigation at UA.

The first initiative, the Digital Engineering Factory (DEF), is a digital engineering environment under development at UA that will be used to support systems and software engineering students. The DEF will host multiple engineering tools, integrated in a 'hub-and-spoke' arrangement with the Violet tool at the 'hub' (Gregory & Salado, 2024a). The goal of the DEF is to provide an environment in which students can access the tools required for a particular engineering course, generate models and data as part of that course, and then transfer that data into another tool (via the hub) to support another engineering course. In its first iteration, the DEF shall support seven UA engineering courses. Each course operates within a set of relevant engineering domains, and this requires a set of relevant engineering tools to support these domains. The planned architecture of the DEF is presented in Figure 2 and represents the array of engineering domains that DEF shall support. The DEF shall also be used to support grading. Students will be able to integrate project data into Violet and generate an RDF representation that can be validated against the UAOS. In this sense, the UAOS shall also be used to promote good engineering practice among students.

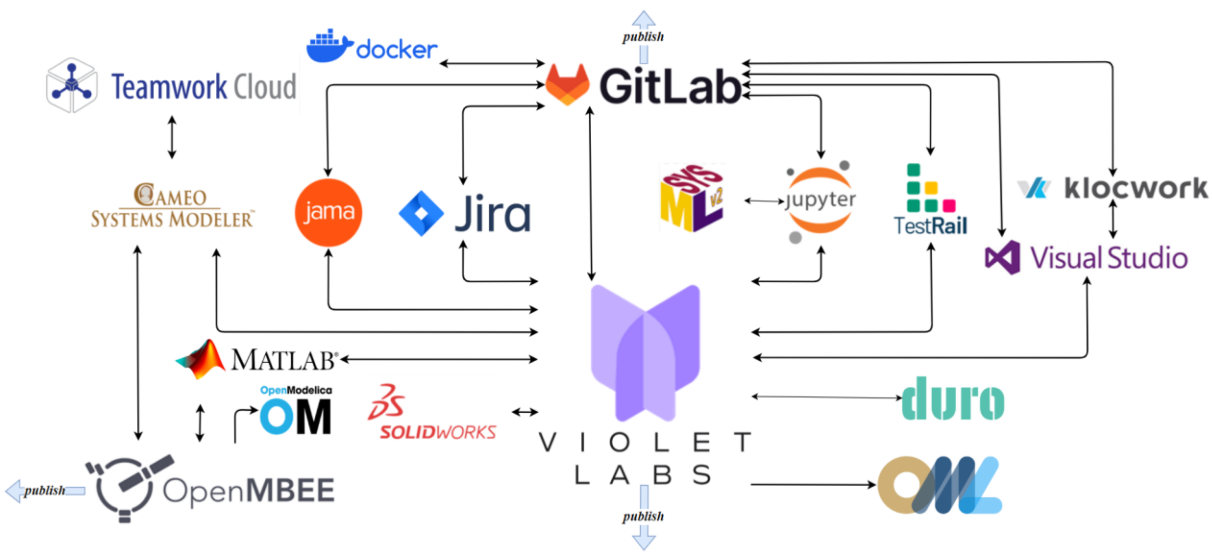

Fig. 2. Software architecture of the Digital Engineering Factory (DEF)

The second initiative is a research project underway at UA that concerns digital test and evaluation. In their 2018 review of the status of model-based systems engineering (MBSE), Madni et al. stated that "model-based test and evaluation is a fertile area of research" (Madni & Sievers, 2018). They also state that "MBSE can guide test planners in developing test strategies and detailed test designs for evaluating technical performance parameters. As tests are performed and data collected, the physical models can be updated with test data" (Madni & Sievers, 2018). It has also been noted that the digital test and evaluation should use "digitally enhanced processes, not just digitized forms of current processes" (Kraft, 2018). In this sense, the development of a digital test and evaluation framework has the potential to enhance the test and evaluation processes by incorporating other developments in digital engineering. The development of a digital test and evaluation framework is being approached in two ways. The first is the production of an ontological version of the Test and Evaluation Master Plan (TEMP) that is used to support Department of Defense (DoD) test programs. A preliminary version of this digital TEMP has been produced, and is referred to as the dTEMP (Gregory & Salado, 2024d, 2024c). The second approach to digital test and evaluation attempts to leverage Bayesian networks to model verification strategies as belief networks where the confidence in a particular feature of the design increases or decreases as more tests are undertaken (Gregory & Salado, 2023a, 2024b). Both the DEF and digital test and evaluation initiatives require representation by a suitable set of domain ontologies.

University of Arizona Ontology Stack

According to Gruber, ontologies play a central role in enabling reusability and interoperability between domains (Gruber, 1991, 1993). Federating the conceptualization of a domain across upper- and lower-level ontologies improves the reusability of higher-level terminology in other domains, and therefore improves interoperability between them. The top-level ontology defines only the most general classes and relations (ISO/IEC 21838-1, 2021) and serves to unify the lower-level core and domain ontologies (Orellana & Mandrick, 2019). Core ontologies extend the top-level ontology to describe more specific classes, such as information and events, but are still applicable to a wide range of domains (Rudnicki, 2019). Domain ontologies formalize the terms that are used in a discipline. Libraries of standards instances can also be used. The UAOS adheres to this structure. It is presented in Figure 3. These ontologies define allowable relations (e.g., a test can verify a requirement), required relations (e.g., a requirement must be verified by a verification activity), and rules of inference (e.g., if Component A contains Component B, and Component B contains Component C, then Component A contains Component C). In this way, complete relation chains that traverse entities sourced from multiple tools can be observed, reasoned upon, and validated. For example, students can use the validation features of the DEF to check whether the requirements that they have produced meet the rules of the ontology – i.e., does each requirement have a name, an ID, and a textual description? Is each requirement satisfied by some aspect of the design? Is each requirement verified by some verification activity? These questions can be answered either by validating against the UAOS or by querying the data.

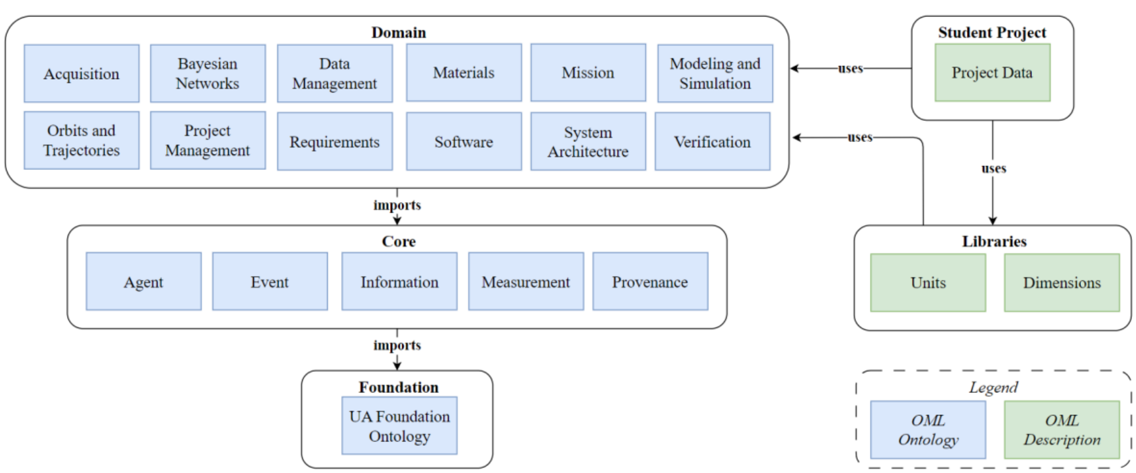

Fig. 3. The University of Arizona Ontology Stack (UAOS)

Ontological Modeling Language

The UAOS has been written in the Ontological Modeling Language (OML) (OpenCAESAR, 2023b). OML is an extension to the Description Logic profile of the Web Ontology Language (OWL 2 DL) and enables reification of relations and convenient use of mixin classes (as aspects). OWL 2 DL, and therefore OML, offers less expressiveness than OWL 2 Full but has been designed to be fully decidable. Data can be then captured in OML descriptions. OML also uses the concept of 'bundles' to allow the modeler to group together multiple descriptions into a closed-world dataset that can be reasoned upon, validated and queried. The closed-world assumption allows the reasoner to highlight the absence of expected information. From these OML descriptions, the OML Rosetta tool can build a triple database in RDF format (OpenCAESAR, 2023a). This enables the use of SPARQL, an RDF query language, to query across the entire dataset (World Wide Web Consortium, 2013). The Semantic Web Rule Language (SWRL) is an extension of OWL and allows for the expression of rules in OWL ontologies (Horrocks et al., 2004). OML Rosetta is an Eclipse-based tool that enables modelers to develop OML ontologies and descriptions. OML Rosetta has been used to support this work because of these built-in validation and querying capabilities (Gregory & Salado, 2024c). Note that in most cases where an existing ontology has been reused, it was provided in OWL and so it was necessary to reproduce it as an OML ontology.

Top-Level Ontology

The UA Foundation Ontology is essentially a simplified OML representation of the Basic Formal Ontology (BFO) and is presented in Figure 4. The BFO is a top-level ontology that is "designed for use in supporting information retrieval, analysis and integration in scientific and other domains" (Arp et al., 2015). Alternative top-level ontologies were considered before BFO was selected. Partridge et. al's 'Survey of top-level ontologies' (Partridge et al., 2020) was referenced during this process. The BFO is grounded in realism, and therefore attempts to categorize the reality of the world rather than human perception of the world. This distinguishes it from some other top-level ontologies, such as DOLCE, which models entities as cognitive artifacts. Ultimately, BFO is an established and well-documented top-level ontology.

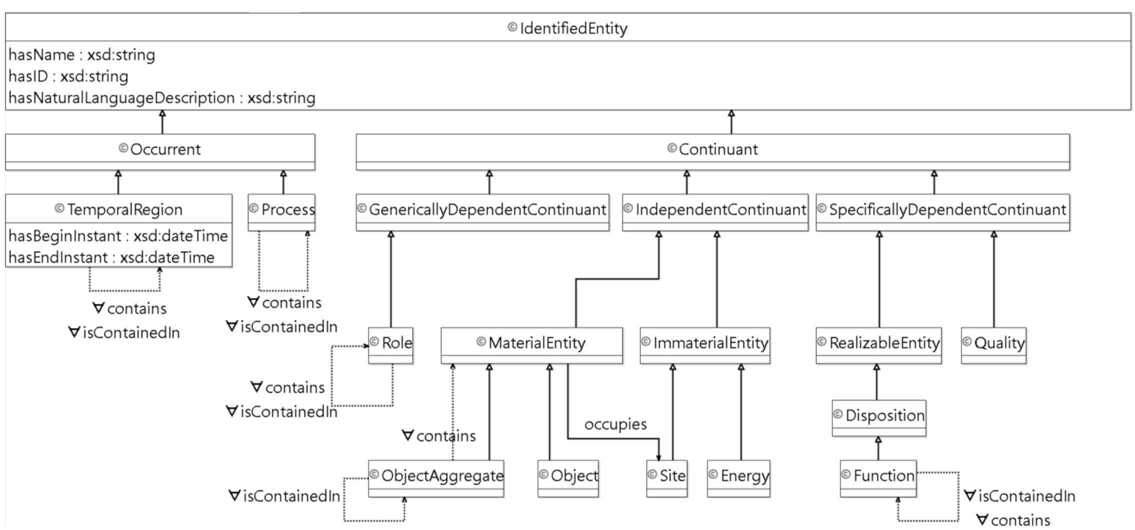

Fig. 4. Partial OML representation of the Basic Formal Ontology (BFO), chosen to be the UA Foundation Ontology

Core Ontologies

Selection of the BFO as the top-level ontology enabled reuse of multiple existing core ontologies. The main contributor to the set of core ontologies was the collection of Common Core Ontologies (CCO) developed by Calspan-University at Buffalo Research Center (CUBRC) (Rudnicki, 2019). All core ontologies except 'Provenance' are based on the CCO. The CCO provide definitions and modeling patterns for crucial, domain-neutral classes and relations. Examples include the measurement of a quantity, and the relation between information, the syntax in which it is represented, and the artifact thar bears the information. The 'Provenance' ontology is based on the W3C PROV-N notation (Moreau et al., 2013) and defines relations crucial for data management such as 'generatedBy' and 'approvedBy'.

Domain Ontologies

Throughout development of the UAOS, the goal has been to base all ontologies on existing ontologies wherever possible. Where this was not possible, the goal was to produce a new ontology based on an accepted domain standard. Multiple UAOS domain ontologies have been based on existing ontologies. The Project Management Ontology is based on PROMONT, a project management ontology (Abels et al., 2006). The Orbits and Trajectories Ontology is based on the Space Object Ontology (Cox et al., 2016). OpenCAESAR, the initiative responsible for the development of OML and OML Rosetta have published multiple OML ontologies as part of their Integrated Model-Centric Engineering (IMCE) project (Elaasar et al., 2023). The UAOS Mission Ontology is based on OpenCAESAR's Mission Ontology, one of these IMCE ontologies. The Bayesian Networks Ontology has been built in accordance with the approach proposed by Fenz in (Fenz, 2012; Fenz et al., 2009). Other UAOS domain ontologies are based on standards. The System Architecture Ontology, for example, is based on ISO 42010 (ISO/IEC/IEEE, 2011). As noted previously, the objective has not been to produce a comprehensive ontological representation of this standard, but rather to capture the required terms and relations in a way that adheres to the standard. The Verification Ontology uses some of the concepts defined in the Virtual Verification of Designs against Requirements (VVDR) methodology (Buffoni et al., 2017; Mengist et al., 2021). There are, however, multiple domain ontologies that are currently not fully developed and are not based on any accepted ontology or standard. This does not necessarily imply that the existence of such ontologies and standards that may support these ontologies do not exist, but rather that we (the authors) have not yet come to fully define them. The Requirements Ontology, for example, currently defines simple requirement types (e.g., performance, physical) and well-known relations (e.g., verify, satisfy), but is not based on any particular requirements model. In this particular case, we defined the minimum required ontology to support the requirements modeling needed to support the DEF and Model-Based Test and Evaluation (MBTE). For all use cases, a simple set of requirement types and the definition of key relations was enough. This is also the case for five other domain ontologies: Acquisition, Data Management, Materials, Modeling and Simulation, and Software. These domain ontologies contain terms and relations that were required to support other domain ontologies but were not in themselves crucial domains. In each of these cases, a 'just enough' approach to development was implemented. Future work will focus on the development of these ontologies so that they incorporate a more comprehensive and theoretically sound definition of their domain. The Requirements Ontology in particular is a priority. A summary of all UAOS ontologies is presented in Table 1.

Table 1: List of ontologies contained in the UAOS

Name	Source	Level	Description	Basis
UA Foundation	n/a	Top-level	Defines fundamental terminology required by all domains (e.g., object, process, role, quality)	BFO (Arp et al., 2015; ISO, 2021)
Agent	n/a	Core	Defines the terminology associated with agency and the roles that agents may undertake (e.g., process owner, agent, organization).	CCO (Rudnicki, 2019)
Event	n/a	Core	Defines the relations associated with events (e.g., manifestation, participation, location in terms of space and time).	CCO (Rudnicki, 2019)
Information	n/a	Core	Defines the terminology associated with information and information artifacts (e.g., descriptive information entity, model).	CCO (Rudnicki, 2019)
Measurement	n/a	Core	Defines the relations between measurements and the quantities that they describe (e.g., measurement, unit, 'has value').	CCO (Rudnicki, 2019)
Provenance	n/a	Core	Defines the relations associated with metadata (e.g., generation, ownership, approval).	PROV-N (Moreau et al., 2013)
Acquisition	MBTE	Domain	Defines acquisition in a way that supports acquisition test programs	n/a
Bayesian Networks	MBTE	Domain	Defines Bayesian relations across networks to support generation of Bayesian networks	Fenz (Fenz, 2012; Fenz et al., 2009)
Data Management	DEF	Domain	Defines information to support version control and repository commitment	n/a
Materials	DEF	Domain	Defines material properties in a way that supports physical architecture definitions	n/a
Mission	DEF	Domain	Defines mission characteristics and relations where a system is designed to participate in a particular mission	OpenCAESAR (Elaasar et al., 2023)
Modeling and Simulation	DEF	Domain	Defines modeling and simulation in terms of inputs, scenarios, and results	n/a
Orbits and Trajectories	DEF	Domain	Defines orbits and trajectories of space objects and their properties	Space Object Ontology (Cox et al., 2016)
Project Management	DEF	Domain	Defines information related to project management such as the allocation of tasks to individuals and teams	PROMONT (Abels et al., 2006)
Requirements	DEF / MBTE	Domain	Defines types of requirements and how they may be derived, refined, satisfied, and verified	n/a
Software	DEF	Domain	Defines software in terms of its version and vendor, and its relations with other software	n/a
System Architecture	DEF / MBTE	Domain	Defines functional and physical architectures of systems in terms of components, interfaces, and hierarchy	ISO 42010 (ISO/IEC/IEEE, 2011)
Verification	DEF / MBTE	Domain	Defines types of verification activities and their relations with plans, procedures, and the agents that undertake them	vVDR (Buffoni et al., 2017; Mengist et al., 2021)

It should also be noted that some of the domain ontologies are dependent on other domain ontologies. The Verification Ontology, for example, is dependent on the Requirements Ontology, and the Mission Ontology is dependent on the System Architecture Ontology. The System Architecture Ontology is presented in Figure 5. In this graphical representation of the ontology, grey boxes represent terms that have been imported from other ontologies. The following ontologies have been referenced:

- *foundation: UA* Foundation Ontology (based on BFO),
- *info:* Information Ontology.

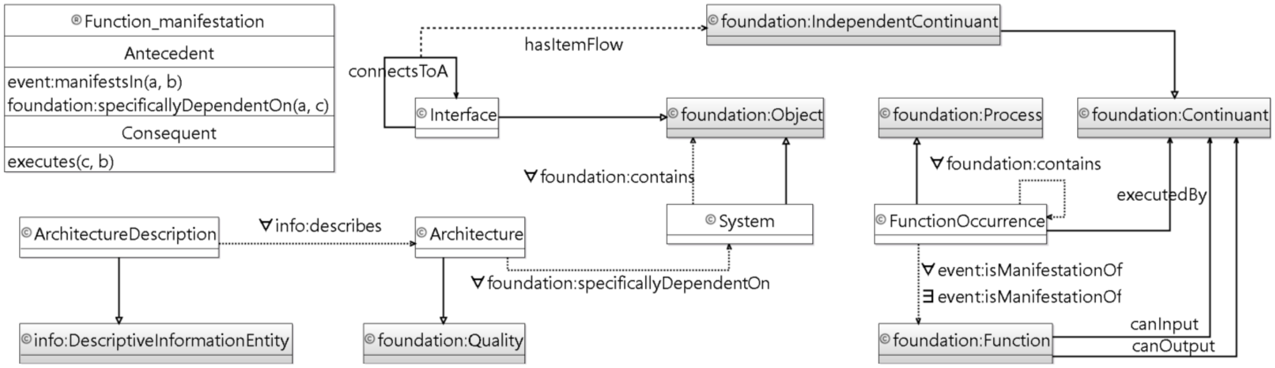

Fig. 5. Partial OML representation of the System Architecture Ontology

White boxes represent terms defined in the System Architecture Ontology. Imported relations have the syntax *RelationSource:RelationName*, where the 'RelationSource' is an imported ontology. Relations defined in the System Architecture Ontology have the syntax *RelationName*. Relations that are unlabeled and denoted as a solid line represent the *rdfs:subClassOf*. Existential and universal operators are also in use:

- if a $\exists$r b:
  - For every instance of **a**, there exists some relation **r** with range **b**.
- if a $\forall$r b:
  - Every relation **r** that has domain **a** has range **b**.

The System Architecture Ontology also contains a SWRL rule through which new information can be inferred from existing information. In this case, if some function *a* manifests in some function occurrence *b*, and that function *a* is a function of some continuant *c*, then it can be inferred that *c* executes *b*.

Applications

The UAOS is a modular ontology stack that has been designed to support multiple use cases. It is therefore not expected that any given project would necessitate the use of all available ontologies. In this section, three applications of the UAOS to a project are described. The domain ontologies used by each project is presented in Table 2. The purpose of the overview presented here is to demonstrate how the modular nature of the UAOS provides flexibility in terms of the projects that the UAOS can support, and to review the effectiveness of these ontologies in capturing the relevant information.

The 'Rover Group Design Project' was a notional student project in which a notional group of three students designed and verified a simple Mars rover (Gregory & Salado, 2024e). This application was designed to represent a typical DEF use case. The students were allocated tasks that included the definition of four system requirements, the development of a suitable Mars rover architecture (i.e., the identification of suitable subsystems), and the completion of various inspections and tests on the system design to verify the requirements. It was possible to capture and model all of the required information using the five domain ontologies specified in Table 2.

Table 2: UAOS domain ontologies used in three example projects

Domain Ontology	Rover Group Design Project	Orbital Analysis Project	Bayesian Verification Project
Acquisition			
Bayesian Networks			✓
Data Management			
Materials			
Mission		✓	
Modeling and Simulation		✓	
Orbits and Trajectories		✓	
Project Management	✓	✓	
Requirements	✓	✓	✓
Software	✓	✓	✓
System Architecture	✓	✓	✓
Verification	✓	✓	✓

The ‘Orbital Analysis Project’ was a similar notional student project. In this case, as well as similar information concerning the project itself and the spacecraft architecture, an analysis of the spacecraft orbit was required to determine whether the batteries onboard were sufficient to provide power to the spacecraft during eclipse (Gregory et al., 2024). An additional three domain ontologies were required to support this analysis: the Mission Ontology, Modeling and Simulation Ontology, and the Orbits and Trajectories Ontology. Together, these eight domain ontologies were sufficient to capture the data required to model this analysis project.

The ‘Bayesian Verification Project’ concerned the verification strategy of a notional Attitude Determination and Control System (ADCS) (Gregory & Salado, 2024b). Relevant requirements were captured using the Requirements Ontology and the architecture of the ADCS itself was captured using the System Architecture Ontology. The Verification Ontology captured the overall test strategy and the relations between the ADCS and individual tests. In this example, the Bayesian Networks Ontology was used to represent existing relations as Bayesian parent-child relations so that a corresponding Bayesian network could be generated.

In all of these examples, the Software Ontology was used so that the source of each entity in the graph database could be captured. (e.g., ‘requirement1 *hasEntitySource* Jama’).

Discussion and Future Work

The UAOS, and the digital engineering projects that it intends to support, are works in progress. Further work is required to explore the application of the UAOS in relevant contexts, and limitations with the current approach been identified.

The domain ontologies have been applied to several relatively simple applications involving notional projects. To be able to capture a more comprehensive project dataset, the domain ontologies themselves may need to be further developed. While the domain ontologies have been developed to accurately reflect the appropriate standard where applicable, they are not necessarily complete representations of that standard. For example, the System Architecture Ontology, displayed in Figure 5, conforms to ISO 42010 (ISO/IEC/IEEE, 2011) but does not fully encapsulate it. This has proven to be an adequate approach for the relatively simple project datasets captured so far, but further development is likely to be required.

Similarly, some domain ontologies (e.g., the Requirements Ontology) have yet to be developed in accordance with an existing standard or ontology. A high-priority action is to explore the relevant literature to identify suitable standards and/or BFO-compliant ontologies that can be used to model these domains. Other relevant ontology developments will be reviewed. The Industrial Ontology Foundry (IOF) is developing multiple BFO-compliant open-source reference ontologies to support the manufacturing and engineering industry (IOF, 2022). OpenCAESAR, the initiative responsible for the development of OML, is developing a suite of IMCE ontologies. We have referenced the IMCE Mission ontology, and we will continue to review the development of other IMCE ontologies. Other domain-specific efforts have been identified, such as the BFO-compliant ‘Ontology of Systems Engineering’ (Orellana & Mandrick, 2019) and the ‘Ontology for Systems Engineering’ (Van Ruijven, 2013). These focus on systems engineering processes in accordance with ISO 15288 (Systems and software engineering – System life cycle processes) (ISO et al., 2015) rather than system architecture, and will be reviewed during the next phase of development.

Beyond the development of those domain ontologies that have already been identified as integral to existing digital engineering initiatives at UA, there may exist other domains that could be incorporated into the UAOS to enable relevant research. One example would be the manufacturing domain, in which other research groups are already undertaking interesting work on ontology-based knowledge representation of the domain (Sanfilippo et al., 2019). Another example would be to incorporate ongoing work in the field of digital engineering (McDermott et al., 2019). To support students specifically, future investigations may consider the implementation of ontologies that incorporate the fields of electrical engineering and mechanical engineering, for example.

The selection of the BFO as the top-level ontology for this work has been justified by the applications so far. Selection of the BFO has enabled the use of the CCO. The CCO have been crucial in providing a solid foundation on which the domain ontologies can be built. Due to the large volume of documentation on the BFO and the CCO, categorization of domain terminology under the relevant core- or foundation-level terms has been achieved. Work within the IOF continues to explore the application of the BFO to industrial engineering cases – this will be crucial as we explore how the BFO can be applied to simulation.

Similarly, the selection of OML rather than OWL 2 DL as the ontology language in which the UAOS has been modeled has been found to be a good approach. The textual interface of OML Rosetta has been a convenient alternative to Protégé, although this decision has necessitated the manual translation of existing ontologies (e.g., BFO) into OML. The UA Foundation Ontology has been named as such as it has not been proven to be a totally complete and accurate representation of the BFO. Future work will aim to complete this translation of these ontologies to OML and validate them against their OWL counterparts.

Each of the applications summarized in this paper have served more as a proof-of-concept than a detailed investigation of the applicability of this approach to real projects. To provide a more rigorous assessment of the UAOS and the semantic web technologies that are enabled by it, more detailed case studies are required that apply the UAOS to authentic project data and attempt to efficiently categorize and reason over the dataset. The required authenticity of this project data can be achieved by generating it through real student projects. Such case studies will include measures to quantify the benefits offered by the deployment of semantic web technologies against the cost of implementation.

Conclusion

In this paper, we have presented the University of Arizona Ontology Stack (UAOS) – a layered, modular ontology stack to support digital engineering initiatives at the University of Arizona. The design science research methodology has been deployed to iteratively develop the UAOS and to ensure that it remains relevant with regards to the initiatives that it intends to support and is rigorous in terms of its foundation in existing standards and ontologies. The UAOS has been written using the Ontological Modeling Language (OML). It currently comprises 18 ontologies – structured across a foundation, core, and domain layer – and two libraries comprising commonly used dimensions and units. The UAOS conforms to the Basic Formal Ontology (BFO) and makes use of the Common Core Ontologies (CCO). Relevant domains have been identified by exploring the use cases associated with two major digital engineering initiatives underway at the University of Arizona. Wherever possible, these domain ontologies will be based on existing BFO-compliant ontologies or relevant standards. To date, the UAOS has been applied to three notional engineering projects, each of which required a different set of the available ontologies. The UAOS was capable of capturing each project dataset, but further work will investigate the application of the UAOS to a more comprehensive and authentic dataset and develop the ontologies accordingly.

Acknowledgments

This material has been produced using funds provided by the Arizona Technology and Research Initiative Fund and by the U.S. Department of Defense through the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)) and the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) under Contract HQ0034-19-D-0003, TO#0510.

References

Abels, S., Ahlemann, F., Hahn, A., Hausmann, K., & Strickmann, J. (2006). PROMONT--a project management ontology as a reference for virtual project organizations. *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops: Proceedings, Part I*, 813–823.

Arp, R., Smith, B., & Spear, A. D. (2015). *Building ontologies with basic formal ontology*. Mit Press.

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: Finding a balance between artifact and theory. *Journal of the Association for Information Systems*, *19*(5), 3.

Batra, S., & Tyagi, C. (2012). Comparative analysis of relational and graph databases. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(2), 509–512.

Buffoni, L., Pop, A., & Mengist, A. (2017). Traceability and impact analysis in requirement verification. *ACM International Conference Proceeding Series*, 95–98. https://doi.org/10.1145/3158191.3158207

Cox, A. P., Nebelecky, C. K., Rudnicki, R., Tagliaferri, W. A., Crassidis, J. L., & Smith, B. (2016). The Space Object Ontology. *Fusion*.

Dertien, S., & Hastings, W. (2021). The State of Digital Thread. *PTC White Paper*.

Dresch, A., Lacerda, D. P., & Miguel, P. A. C. (2015). A distinctive analysis of case study, action research and design science research. *Revista Brasileira de Gestão de Negócios*, *17*, 1116–1133.

Dunbar, D., Hagedorn, T., Blackburn, M., Dzielski, J., Hespelt, S., Kruse, B., Verma, D., & Yu, Z. (2023). Driving digital engineering integration and interoperability through semantic integration of models with ontologies. *Systems Engineering*, *January*, 1–14. https://doi.org/10.1002/sys.21662

Elaasar, M., Rouquette, N., & Jenkins, S. (2023). *OpenCAESAR Integrated Model-Centric Engineering (IMCE) Vocabularies*. OpenCAESAR. https://github.com/opencaesar/imce-vocabularies

Fenz, S. (2012). An ontology-based approach for constructing Bayesian networks. *Data and Knowledge Engineering*, *73*, 73–88. https://doi.org/10.1016/j.datak.2011.12.001

Fenz, S., Tjoa, A. M., & Hudec, M. (2009). Ontology-based generation of bayesian networks. *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems, CISIS 2009*, *i*, 712–717. https://doi.org/10.1109/CISIS.2009.33

Gregory, J., Nair, M. H., Bullegas, G., & Saaj, M. C. (2021). Using Semantic Systems Engineering Techniques to Verify the Large Aperture Space Telescope Mission - Current Status. *Model Based Space Systems and Software Engineering MBSE2021*.

Gregory, J., & Salado, A. (2023a). Model-Based Verification Strategies Using SysML and Bayesian Networks. *Conference on Systems Engineering Research (CSER)*, 1–11.

Gregory, J., & Salado, A. (2023b). *UA Ontology Stack GitHub Repository*. https://github.com/joegregoryphd/UAOntologyStack

Gregory, J., & Salado, A. (2024a). A Digital Engineering Factory for Students. *Conference on Systems Engineering Research (CSER)*.

Gregory, J., & Salado, A. (2024b). A Semantic Approach to Spacecraft Verification Planning using Bayesian Networks. *IEEE Aerospace Conference*.

Gregory, J., & Salado, A. (2024c). An Ontology-based Digital Test and Evaluation Master Plan (dTEMP) Compliant with DoD Policy. *Systems Engineering*.

Gregory, J., & Salado, A. (2024d). Applying the Digital Test and Evaluation Master Plan (dTEMP) to a Mars Rover Test Program. *Naval Engineers Journal*.

Gregory, J., & Salado, A. (2024e). Implementing a Student Rover Design Exercise in the Digital Engineering Factory. *IEEE Aerospace Conference*.

Gregory, J., Salado, A., Hoag, L., Jones, B., Marmar, M., & Zute, A. (2024). Spacecraft Test and Evaluation using Semantic Web Technologies. *AIAA SciTech*.

Gruber, T. R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, 601–602. http://www.cin.ufpe.br/~mtcfa/files/10.1.1.35.1743.pdf

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

Hevner, A., & Chatterjee, S. (2010). *Design research in information systems. Theory and practice*. Springer.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, *19*(2), 4.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2008). Design science in information systems research. *Management Information Systems Quarterly*, *28*(1), 6.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. In *W3C Member Submission* (Issue May).

IOF. (2022). *Industrial Ontologies Foundry*. https://www.industrialontologies.org/

ISO/IEC/IEEE. (2011). *Systems and software engineering -- Architecture description*.

ISO/IEC 21838-1. (2021). *Information technology — Top-level ontologies (TLO) — Part 1: Requirements*. *2021*, 23. https://www.iso.org/standard/71954.html

ISO. (2021). *ISO 21838-2: Information technology — Top-level ontologies (TLO) — Part 2: Basic Formal Ontology (BFO)* . *2010*.

ISO, IEC, & IEEE. (2015). *ISO/IEC/IEEE 15288:2015(E) - Systems and software engineering - System life cycle processes*.

Kraft, E. M. (2018). A Disruptive Application of Digital Engineering to Optimize Aircraft Developmental Test \& Evaluation. *AIAA 2018 Aviation Systems Conference, Atlanta, GA*.

Madni, A. M., & Sievers, M. (2018). Model-based systems engineering: Motivation, current status, and research opportunities. *Systems Engineering*, *21*(3), 172–190. https://doi.org/10.1002/sys.21438

Maló, P. M. N. (2013). Hub-and-spoke interoperability: An out of the skies approach for large-scale data interoperability.

Universidade Nova de Lisboa, Master's Thesis.

McDermott, T., Collopy, P., Nadolski, M., & Paredis, C. (2019). The Future Exchange of Digital Engineering Data and Models: An Enterprise Systems Analysis. *Conference on Systems Engineering Research (CSER)*, *153*, 260–267. https://doi.org/10.1016/j.procs.2019.05.078

Medhi, S., & Baruah, H. K. (2017). Relational database and graph database: A comparative analysis. *Journal of Process Management and New Technologies*, *5*(2), 1–9.

Mengist, A., Buffoni, L., & Pop, A. (2021). An integrated framework for traceability and impact analysis in requirements verification of cyber–physical systems. *Electronics (Switzerland)*, *10*(8). https://doi.org/10.3390/electronics10080983

Moreau, L., Missier, P., Cheney, J., & Soiland-Reyes, S. (2013). PROV-N: The Provenance Notation. *W3C Recommendation*, *April*, 1–34. http://www.w3.org/TR/2013/REC-prov-n-20130430/

Nabukenya, J. (2012). Combining case study, design science and action research methods for effective collaboration engineering research efforts. *2012 45th Hawaii International Conference on System Sciences*, 343–352.

Noy, N. F., McGuinness, D. L., & others. (2001). *Ontology development 101: A guide to creating your first ontology*.

Olsina, L. (2021). Applicability of a Foundational Ontology to Semantically Enrich the Core and Domain Ontologies. *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K - Proceedings*, *2*(Ic3k), 111–119. https://doi.org/10.5220/0010641000003064

OpenCAESAR. (2023a). *Github - OML Rosetta*. https://github.com/opencaesar/oml-rosetta

OpenCAESAR. (2023b). *Ontological Modeling Language 2.0.0*. https://www.opencaesar.io/oml/

Orellana, D., & Mandrick, W. (2019). The Ontology of Systems Engineering: Towards a Computational Digital Engineering Semantic Framework. *Procedia Computer Science*, *153*, 268–276. https://doi.org/10.1016/j.procs.2019.05.079

Otter, M., Thuy, N., Bouskela, D., Buffoni, L., Elmqvist, H., Fritzson, P., Garro, A., Jardin, A., Olsson, H., Payelleville, M., Schamai, W., Thomas, E., & Tundis, A. (2015). Formal Requirements Modeling for Simulation-Based Verification. *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, *118*, 625–635. https://doi.org/10.3384/ecp15118625

Partridge, C., Mitchell, A., Cook, A., Leal, D., Sullivan, J., & West, M. (2020). *A survey of top-level ontologies: to inform the ontological choices for a Foundation Data Model*. *November*, 143. https://doi.org/10.17863/CAM.58311

Patel, A., & Jain, S. (2021). Present and future of semantic web technologies: a research statement. *International Journal of Computers and Applications*, *43*(5), 413–422.

Pries-Heje, J., Baskerville, R., & Venable, J. R. (2008). *Strategies for design science research evaluation*.

Rudnicki, R. (2019). An Overview of the Common Core Ontologies. *CUBRC Report.*

Rudnicki, R., Smith, B., Malyuta, T., & Mandrick, W. (2016). *Best Practices of Ontology Development*. 1–16.

Salado, A., & Kannan, H. (2019). Elemental patterns of verification strategies. *Systems Engineering*, *22*(5), 370–388.

Sanfilippo, E., Belkadi, F., & Bernard, A. (2019). Ontology-based knowledge representation for additive manufacturing. *Computers in Industry*, *109*, 182–194.

Van Ruijven, L. C. (2013). Ontology for systems engineering. *Procedia Computer Science*, *16*, 383–392. https://doi.org/10.1016/j.procs.2013.01.040

World Wide Web Consortium. (2013). *SPARQL 1.1 Query Language*. W3C.

Zimmerman, P., Gilbert, T., & Salvatore, F. (2019). Digital engineering transformation across the Department of Defense. *The Journal of Defense Modeling and Simulation*, *16*(4), 325–338.

Biography

Dr. Joe Gregory is a postdoctoral research associate at the University of Arizona. His research interests include digital engineering, model-based systems engineering, and the application of semantic web technologies to support engineering. In 2022, he received his PhD in Aerospace Engineering from the University of Bristol for his development of the SysML-based 'Spacecraft Early Analysis Model'. He is the co-chair of the Digital Engineering Information Exchange (DEIX) Taxonomy Working Group.

Dr. Alejandro Salado has over 15 years of experience as a systems engineer, consultant, researcher, and instructor. He is currently an associate professor of systems engineering with the Department of Systems and Industrial Engineering at the University of Arizona and the director of its systems engineering programs. Dr. Salado holds a BS/MS in electrical and computer engineering from the Polytechnic University of Valencia, a MS in project management and a MS in electronics engineering from the Polytechnic University of Catalonia, the SpaceTech MEng in space systems engineering from the Technical University of Delft, and a PhD in systems engineering from the Stevens Institute of Technology.