Putting the O in Model – Ontologies for MBSE

Paul Hampton BSc (Hons) CEng MBCS CITP, Safety Critical Systems Club, paul.hampton@scsc.uk
Stephen Powley MEng MIET MINCOSE, Coventry University, stephen@omflow.com
Dave Banham BEng CEng MIET, BlackBerry QNX, dbanham@qnx.com

Categorisation

- Accessibility: BEGINNER
- Application: GOOD PRACTICE
- Topics: MBSE, Ontology, Risk

Abstract

With origins in antiquity and ancient philosophers speculating on the nature of being, “ontology” and “ontological modelling” perhaps have a reputation for being esoteric, impenetrable and only for the purist of academics. In the last couple of decades, however, great work has been done to make ontological concepts and models readily accessible, along with the essential toolkits required to apply them at a practical level.

During system development, and especially with Model-Based Systems Engineering (MBSE), it is essential to conceptualise the world that the system will inhabit, and this involves developing and structuring concepts, terms and their relationships. An ontology is an explicit specification of a conceptualisation and provides the foundation for MBSE and architecture descriptions. Stronger foundations support longer-lived models that work better together. Increasingly, ontologies are also being recognised as essential for enterprise knowledge systems, interoperable standards and explainable Artificial Intelligence (AI).

This paper, from the Safety Critical Systems Club (SCSC) Ontology Working Group (OWG), introduces modelling with ontologies. Examples show different ways to model aircraft flights and how this can affect interoperability. The examples use the Unified Foundation Ontology (UFO) and the OntoUML modelling language based on UFO and UML. The authors consider the importance of different perspectives and capturing these in our modelling and show the types of challenge a top-level ontology can help with. This demonstrates how relevant and important ontologies are for enhancing collaboration and understanding between stakeholders and the models they use.

The paper concludes by outlining the OWG’s objectives for developing ontological models that will benefit everyone working in domains that involve understanding risk, such as safety and security.

Introduction

An ontology is commonly perceived as some esoteric formalisation, which is nice to have, but not relevant or accessible to the day-to-day work of an individual working on real systems. Ontologies are, however, everything to do with a systems engineer’s day-to-day work. When developing a system, engineers are modelling the world and this involves developing concepts, terms and their relationships – which can be formalised as an ontology, or “an explicit specification of a conceptualization” [Gruber 1993]. If not explicitly specified, establishing a shared understanding of interrelated concepts will be much harder and miscommunication may be more common.

As Artificial Intelligence (AI) becomes ever more present in our lives, ontologies are increasingly in focus to help address challenges of AI trustworthiness and explainability. Ontologies can serve as contracts of meaning, providing a shared logical representation of what things mean and the way that we see the world that is interpretable by both humans and machines. Importantly, ontologies constrain the kinds of things recognised by a worldview, how they are related, and what is logically entailed when an assertion is made. These are essential features for engineers looking to integrate AI into safety-critical modelling processes or to train AI to work with their models.

The lack of a shared ontology (or a bad ontology) can lead to poor designs that introduce additional complexity, ambiguity and inconsistency. In fact, this type of issue can lead to serious real-world consequences when the deployed system is difficult to explain or diverges from the real-world intended behaviour. One such issue occurred in 2023 when interoperability issues between air navigation systems led to an outage of the UK air traffic control services (NATS). Ultimately, the issue arose from the failure of a system to correctly handle a particular occurrence of duplicate waypoint identifiers in flight planning data. A waypoint is a geographic position denoted by latitude and longitude values and assigned an identification code. A series of identifiers is used in flight planning to help define the route an aircraft intends to follow. Duplicate waypoint identifiers are well known, in fact there are over 3000 worldwide. In this case, a particularly rare combination of circumstances and data (the first occurrence in 15 million flight plans) caused a critical exception in the processing logic that took the system out of service for many hours. The disruption affected 700,000 passengers with a total cost impact, estimated by the CAA, of at least £75m [Halliwell et al. 2024]. The NATS outage could have been avoided in a world with a globally shared ontology that enabled reasoning. As the aviation industry has evolved more organically over time from local air traffic concerns to global ones, it is understandable that duplicates may have arisen and that it is now difficult to reimpose a stricter model. A system that incorporated ontology-based reasoning could allow the system to infer from other related information that the duplicate identifiers in fact refer to different locations and thus prevent the processing from arriving at an illogical flight plan. The corollary to this is that if you are developing a complex system of systems then the risk of unexpected failure will be mitigated by having a good ontological model.

The Safety Critical Systems Club (SCSC) is a global professional network for sharing knowledge about system safety and supports the community in developing best practice guidance through its working group collaborations. The Club has found that the natural language used in standards and other formal documents lacks the formalisation and structure that ontologies bring, and this typically results in ambiguities. These can arise through inconsistent use, and sometimes misuse, of domain terms that hinder the reader's and document maintainer's clear understanding of the original author's intent. Such issues then seep into the broader business context where different groups of people collaborate, for example in customer facing roles, business- to-business interactions, consortiums, onboarding new staff, and working with consultants. An emerging area of guidance for the Club is therefore to champion ontologies as a means of improving how people collaborate, articulate designs and ultimately arrive at better engineered and hence safer systems. In ongoing work, SCSC's Ontology Working Group is developing an ontology to support risk-based decision making and data safety.

Aims

This paper introduces ontological modelling in a familiar context for systems engineers and discusses why this is important. Some strategies for addressing the challenges of developing explicit, shared conceptualisations for safety- and security-critical engineering are also considered. The example applies a top-level ontology to illustrate the type of interoperability challenge that can go unnoticed if this careful conceptual modelling is neglected. The authors seek to convey the importance of capturing different perspectives and accounting for these in our modelling approaches, and to show why the growing complexity of modern systems means now is the right time to put the O in model.

Approach

Initially the OWG researched models of risk that other groups and organisations had developed to see if these were appropriate for adoption. These included the Threat & Risk Conceptual Model & Ontology (TRM) [Casanave 2018] and the ISO 31000 risk management guidelines [ISO 2018]. The group also explored ontological frameworks such as the Semantic Modelling for Information Federation (SMIF) [OMG 2016], ISO/IEC 21838 Basic Formal Ontology (BFO) [ISO 2021] and the Unified Foundation Ontology (UFO) [Guizzardi et al. 2022]. The group found UFO to be the most promising ontological framework for the purpose of modelling risk and will monitor, synthesise and adapt other works to provide a suitable model for risk and value.

The idea of modelling concepts and relationships is not new and familiar languages such as the Unified Modelling Language (UML) [OMG 2017] and its Systems Engineering (SE) customisation, SysML [OMG 2019], can be used for this purpose. Both can be too general for many modelling tasks and, without a framework to impose some constraints, it is easy to model things that are inconsistent or ambiguous. This is why the OWG use the **Unified Foundation Ontology (UFO)** and **OntoUML**, a domain-specific language (DSL) tailored for ontological modelling in UML based on UFO[1] [Marek Suchánek 2018][Guizzardi 2021]. UFO comes in three parts: a top-level for fundamental "things" (UFO-A), a mid-level for "events" (UFO-B), and an extension for "social" aspects (UFO-C). UFO-A is the starting point for modelling as it details the most fundamental concepts, some of which are introduced below.

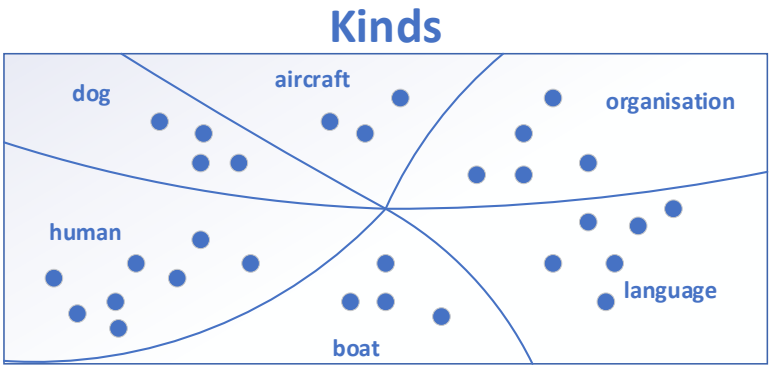

Figure 1 - UFO Kinds (diagram inspired by [Guizzardi 2021])

[1] OntoUML is a customisation of UML with software support via application plugins for Visual Paradigm and Enterprise Architect (deprecated)

Figure 1 illustrates the core concept of a UFO **Kind**. Each 'dot' ● in the diagram indicates an individual instance of a particular **Kind**. This concept reflects everyday real-world objects like people, aircraft and animals. **Kind**s are objects that endure for their entire lifetime with essential identifying characteristics that all their instances must possess (ontologists call this an identity principle). It is possible to distinguish between instances of **Kind**s: for example, different people have different identities based on properties commonly used to distinguish people (e.g. eye-colour, place of birth, genes).

It is also useful to be able to express the **Roles** and **Phases** that **Kinds** might have during their lifetime. For example, a human may be a student for some part of their life (a **Role**), but will cease to be a student when they leave education. Similarly, a puppy is a **Phase** a dog will go through, which is transitory and only for the early part of their life. The following diagram shows how particular instances may exist in a particular **Phase** or **Role** at some point in their lifetime. For example: an organisation may be dissolved at some point; some languages are no longer spoken as a native language by a community of people so are considered "dead"; a boat may at some point in its lifetime be used for hospitality (e.g. parties, ceremonies, etc.)

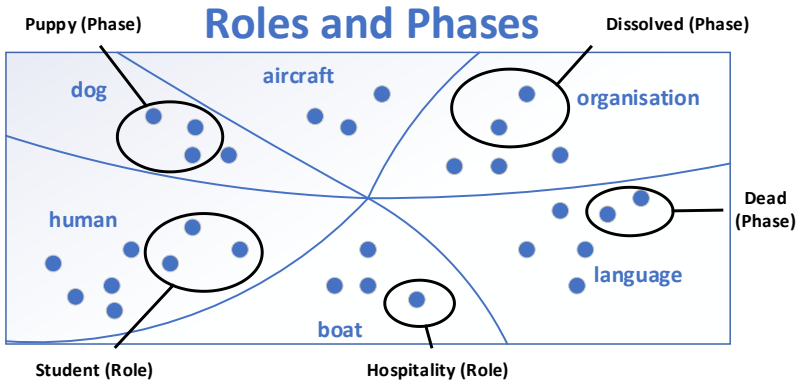

Figure 2 - UFO Roles and Phases

Another concept that is useful when modelling the world is that of a **Mixin**, used to describe types based on shared properties of objects that do not all belong to the same **Kind**; in other words a **Mixin** spans multiple **Kind**s that do not share the same identity principle (a red boat does not have red hair colour, but both are red things). The example shows a **Mixin** that includes **Kind**s that can be insured.

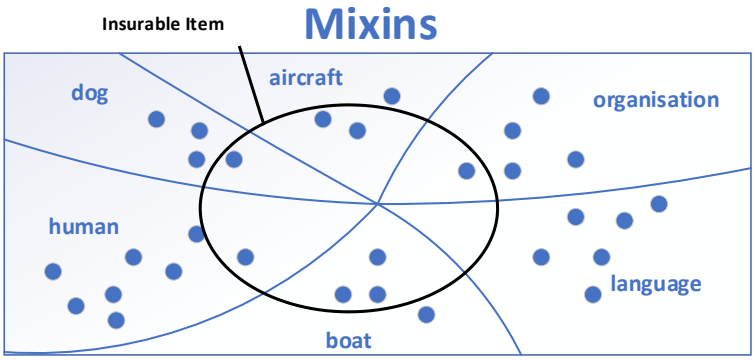

Figure 3 - UFO Mixins

These concepts, along with the associated rules for their application defined by OntoUML, give a powerful and consistent toolkit for describing important concepts and relationships in our world.

Modelling with OntoUML

In the NATS example presented earlier, a flight was represented as a series of waypoints that could have duplicate identifiers. Here we explore some ways to build richer ontological models of flights that consider factors other than waypoints.

Figure 4 shows an example of an OntoUML model for part of an aviation system of systems. In the diagram, boxes (which in UML represent *classes)* show 'things' that are present in an individual's view of the world. The ontological aspects are expressed as stereotypes (that is the part of the class definition quoted with Guillemets, like this: «role»). Classes with the same stereotype share the same colour in the diagram. Lines between the boxes represent the relationships between classes. Arrows with a filled white triangle are *generalizations*, or in other words, one 'thing' subtypes the other 'thing' that is pointed to (for example, a **Passenger** is a subtype of **Human**). An open arrow relationship shows a looser *association*[2] between things (note that *associations* also have stereotypes). Expressions at the end of each association, such as **0..*** and **1..*** (read as *zero or more* and *one or more* respectively) show *multiplicity* – the range of the number of each type of thing that can be related. ***Insurable Item*** appears on the diagram in italics – this indicates it is an abstract class, meaning it cannot have direct instances.

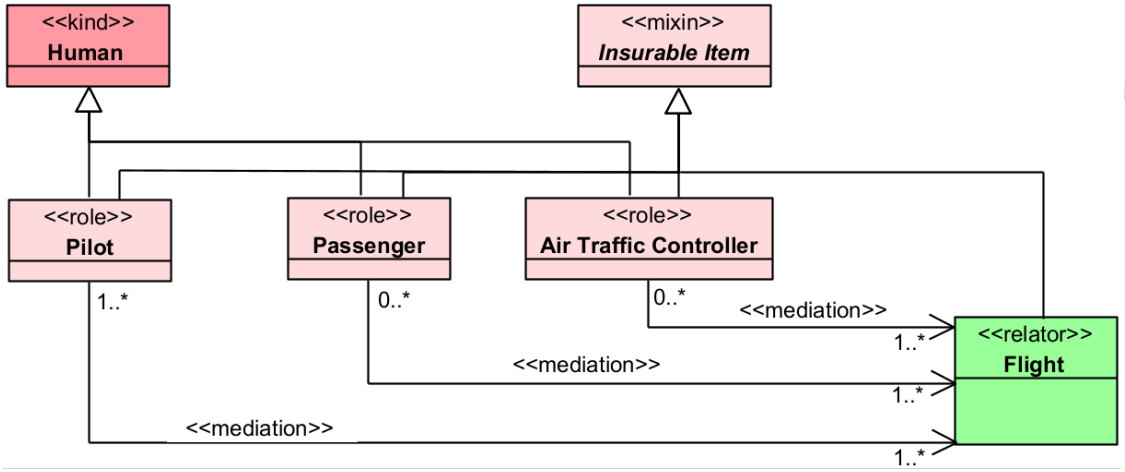

Figure 4 – Example of a UFO model of an aviation system of systems using OntoUML

A **Flight** always has one or more **Pilot,** but maybe more, and **Pilot**s may be involved in one or more **Flight**s. A particular **Flight** may have zero or more **Passenger**s, with none if the **Pilot** is the only occupant. A **Human** may also be a **Passenger** on more than one **Flight**. This model is under-constrained as it does not preclude, for example, **Pilot**s from being **Passenger**s on the same flight, or being **Pilot** on different aircraft simultaneously. This type of issue requires additional constraint logic, such as described by the OntoUML anti-pattern catalogue [Prince Sales, Tiago 2014]. A **Human** is not an ***Insurable Item*** unless they are have a certain role. This might be useful if uninsurable risks need to be considered, but not if a subset of **Pilot**s are uninsurable, e.g. they are unlicensed. In that case, additional insurable subclasses of **Pilot** could be added. **Flight**s are also ***Insurable Item***s, but the model could be under-constrained. A **Flight** by an unlicensed **Pilot** without an **Air Traffic Controller** would still be considered an ***Insurable Item*** according to Figure 4, but perhaps not by an insurer!

[2] OntoUML redefines the meaning of the UML navigability arrow to indicate the direction of a semantic relationship.

Different Perspectives

The concept of a **Flight** will seem familiar to anybody who has caught a plane to go on holiday. In Figure 4, the **Flight** only exists as a relationship between a **Pilot**, and maybe the **Air Traffic Controller** and **Passenger**s – indicated by the «relator» stereotype. Relators provide «mediation» between the things they depend on for their existence. Here, there is no flight without a **Pilot**. **Passenger**s are often on the **Flight** (but not always) and at least one **Air Traffic Controller** usually helps with navigation (but not always). This perspective might provide a useful way of modelling flights at a small private airfield, where **Flight**s are created on an ad hoc basis when a **Pilot** and **Passenger**s are in the plane and want clearance to take off. There might be no prior concept of an actual flight until that moment. In this way of modelling, if a **Flight** never took place, it never existed. So maybe this is not the same type of thing as the **Flight** you took to your holiday destination after all?

In Figure 5, **Flight** is considered a type of thing that has its own independent existence, not one that relies on relationships to exist. This is probably closer to the concept you imagined going on holiday. The **Pilot**'s relationship with the **Flight** is expressed in their ownership of a **Pilot License** to give them authority to be the **Pilot** of an aircraft. **Aviation Authority** is a «subkind» because its instances maintain their identity as **Aviation Authority** all the time they exist (in contrast to «role» and «phase», which are transitory subtypes of «kind»). This perspective could better represent **Flight**s by a commercial transport airline. In this situation, **Flight**s are planned months in advance and might take place, be delayed or be cancelled. The cancelled **Flight** still has relevancy for operational needs, such as charges to airlines for airport services or payment of passenger compensation. The purely relational approach of the first diagram would not support this permanency of a **Flight** if it never took place.

The representation in Figure 5 assumes that all **Human**s are always ***Insurable Item***s. This simplifies the model, but makes it more difficult to represent people doing uninsurable things. For a standalone flight booking system this might be adequate, but not if that system needed to integrate with insurers' models. Ontologies allow stakeholders and modellers to work together to identify such challenges in advance and design compatible systems that will be more resilient when deployed together.

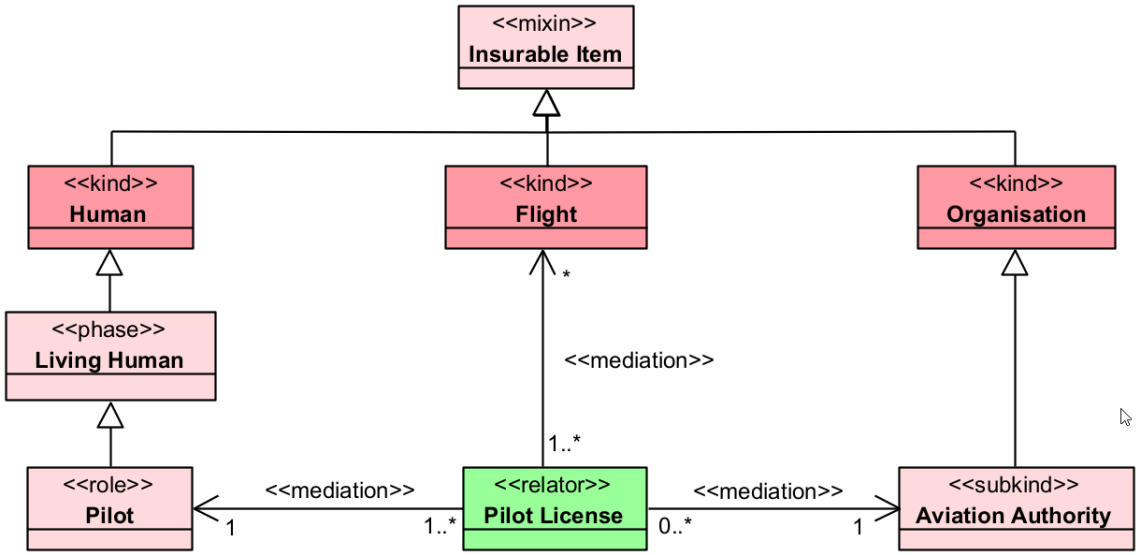

Figure 5 – A Different perspective on modelling an aviation system of systems

Also note that in the first model, a **Human** has identity and is a **Kind**, in the second, the **Living Human** is seen as a **Phase** of a **Human**. It makes intuitive sense that **Pilot**s must be living, but this constraint may be inadequate in isolation. Recall that this ontology is intended to support permanency of **Flight**s, so **Pilot**s will inevitably subsequently die at some point after an instance of a **Flight** has been recorded. This means that temporal considerations would need to also be considered in real-world scenarios.

The above examples show that the concept of a **Flight** can be modelled in very different ways for different purposes, and that they can all be correct at the same time. The importance of modelling different perspectives can also be visualised by considering the shadows cast by the object in Figure 6. Just as with the **Flight** examples, there can be several perspectives of something in the world that are different, while none is incorrect.

Figure 6 – The same, but different © [Furian n.d.]

Discussion

The **Flight** examples illustrate how common situations can be conceptualised in different, but nevertheless correct, ways depending on the context and stakeholder's perspective. Where possible, a shared model and conceptualisation should be developed covering the entire enterprise; this would eliminate logically inconsistent situations from arising, such as in the NATS example, which illustrated the perils of trying to interoperate between uncoordinated models. Perhaps if more rigour had been applied to the formalisation of each jurisdiction's model to address interoperability challenges, the edge case that led to an expensive and potentially life-threatening incident might have been averted.

For systems engineers using MBSE, even with good ontologies there remain engineering challenges to integrating models built on different worldviews. In the **Flight** example, given the two OntoUML models we could provide at least a partial solution using UML *namespaces* to group concepts according to shared context. Then, where it is important to distinguish different concepts of **Flight**, contextual information can be shown in the model. For example, with namespaces 'Private Airfield' and 'Commercial Airline', UML notation would be **Private Airfield::Flight** and **Commercial Airline::Flight**. More advanced contextual information display in modelling tools is becoming possible, supported by standards like OMG's Multiple Vocabulary Facility (MVF) [OMG 2024]. This kind of approach can be valuable when stakeholders cannot agree on a common definition of a term. Accommodating different meanings for a term might add complication to the model, but this can pay dividends if it reduces confusion for the engineers who are implementing the system.

A particularly important area of shared vocabulary across the SCSC working groups is that of risk. Modelling risk is a difficult challenge, but by applying the techniques outlined here the OWG is confident of making a strong contribution. It is difficult to formalise a risk model for an isolated domain (safety, security, etc.) without a more general model for risk itself. There have been many different attempts by other working groups and organisations to model risk, largely because, as The Society for Risk Analysis has shown [SRA 2018], there are many different perspectives and ways of viewing the problem. The OWG has assessed many different approaches but believes the most fruitful approach is through models that consider risk as an experience for participants and take value as a key consideration when thinking about risk [Sales et al. 2018]. For example, any discussion around risk needs to consider how participants value items at risk, and indeed, how those who aim to exploit a

vulnerability value what they will gain versus how they value their liberties that may be lost if they are caught. The OWG aims to develop clear, systematic vocabulary and an unambiguous model of risk and value concepts and their relationships. A particular focus is to provide a common framework for expressing safety and security risks, which is a pressing concern for practitioners in both disciplines as they become increasingly integrated. This will also be of value in helping AI comprehend how humans understand risk in the engineering context, making it a more useful and reliable tool for safety-critical applications.

Conclusion

Whether specified explicitly or not, concepts and the relationships between them are fundamental to any SE activity. Especially for complex, safety-critical systems of systems, which increasingly rely on connected off-the-shelf devices, it is becoming essential that explicit conceptualisations are developed, shared and understood by all stakeholders to avoid ambiguities leading to vulnerabilities. It is hard to overstate the importance of this for the safety/security community, where arguably ontologies must become an essential pillar. Without them, the work of each discipline becomes harder and open to inconsistencies. At best this slows the development of best practice, at worse, it could lead to significant incidents that risk lives and livelihoods. The need has never been greater with emerging technological paradigms defying traditional analysis methods: for example, it is difficult to see how a robust understanding of Artificial Intelligence (AI) decision-making can be developed if the concepts and relationships it operates on are not supported by formal models. The SCSC believes that great benefit will be derived by systems engineers adopting more formal approaches to ontologies. For their part, the SCSC is working to ensure guidance publications all hold a consistent view on the language of risk as applied to risk-based decision making in engineering.

References

[Casanave 2018] Casanave, C. (2018) Threat & Risk Conceptual Model & Ontology. Available at: https://github.com/ModelDriven/ThreatRisk/blob/master/README.md (Accessed: 6 May 2025).

[Peter Furian n.d.] Peter Hermes Furian (no date) *Different Shadows from Same Object Circle Square Triangle Point of View Stock Vector - Illustration of difference, optical* [Digital]. Available at: https://www.dreamstime.com/different-shadows-same-object-circle-square-triangle-point-view-image228893902 (Accessed: 17 August 2025).

[Gruber 1993] Gruber, T.R. (1993) 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, 5(2), pp. 199–220. Available at: https://doi.org/10.1006/knac.1993.1008.

[Guizzardi 2021] Guizzardi, G. (2021) Philosophical Ontology and Domain Modeling - An Introduction to the OntoUML Approach (Parts 1-3). Available at: https://www.youtube.com/watch?v=ENGEIhbnAx4 (Accessed: 9 May 2025).

[Guizzardi et al. 2022] Guizzardi, G., Botti Benevides, A., Fonseca, C.M., Porello, D., Almeida, J.P.A. and Prince Sales, T. (2022) 'UFO: Unified Foundational Ontology', *Applied Ontology*. Edited by S. Borgo, A. Galton, and O. Kutz, 17(1), pp. 167–210. Available at: https://doi.org/10.3233/AO-210256.

[ISO 2018] ISO (2018) *ISO 31000:2018 - Risk management — Guidelines*, *ISO*. Available at: https://www.iso.org/standard/65694.html (Accessed: 9 May 2025).

[ISO 2021] ISO (2021) 'Information technology — Top-level ontologies (TLO) — Part 2: Basic Formal Ontology (BFO)'. ISO. Available at: https://www.iso.org/standard/74572.html (Accessed: 9 May 2025).

[Halliwell et al. 2024] Halliwell, J., Chambers, S., Cropper, P. and Foulsham, M. (2024) *Independent Review of NATS (En Route) Plc's Flight Planning System Failure on 28 August 2023 – Final Report*. Independent Review CAP2993 (Ver. 1). UK Civil Aviation Authority. Available at: https://www.caa.co.uk/our-work/publications/documents/content/cap2993/#

[OMG 2016] OMG (2016) 'Semantic Modeling for Information Federation (SMIF)'. OMG. Available at: http://www.omg.org/spec/SMIF (Accessed: 9 May 2025). [OMG 2017] OMG (2017) 'The Unified Modeling Language Specification Version 2.5.1'. OMG. Available at: https://www.omg.org/spec/UML/ (Accessed: 2 April 2024).

[OMG 2019] OMG (2019) 'About the OMG Systems Modeling Language Specification Version 1.6'. OMG. Available at: https://www.omg.org/spec/SysML/1.6 (Accessed: 2 April 2024).

[OMG 2024] OMG (2024) 'Multiple Vocabulary Facility (MVF), v1.0'. Object Management Group (OMG). Available at: https://www.omg.org/spec/MVF/1.0/PDF (Accessed: 14 October 2024).

[Prince Sales 2014] Prince Sales, Tiago (2014) *RelOver anti-pattern — OntoUML specification documentation*. Available at: https://ontouml.readthedocs.io/en/latest/anti-patterns/RelOver/index.html (Accessed: 24 April 2025).

[Sales et al. 2018] Sales, T.P., Baião, F., Guizzardi, G., Almeida, J.P.A., Guarino, N. and Mylopoulos, J. (2018) 'The Common Ontology of Value and Risk', in J. Trujillo (ed.) *Conceptual Modeling, ER 2018, Lecture Notes in Computer Science*. Cham, Switzerland: Springer. Available at: https://doi.org/10.1007/978-3-030-00847-5_11.

[SRA 2018] SRA (2018) 'The Risk Analysis Glossary from the Society for Risk Analysis'. The Society For Risk Analysis. Available at: https://www.sra.org/risk-analysis-introduction/risk-analysis-glossary/ (Accessed: 12 May 2025).

[Suchánek 2018] Suchánek, M. (2018) *OntoUML specification*, *OntoUML specification*. Available at: https://ontouml.readthedocs.io/en/latest/ (Accessed: 27 March 2025).