Electronic Communications of the EASST
Volume 079 (2020)

Interactive Workshop
on the Industrial Application of Verification and Testing
ETAPS 2020 Workshop
(InterAVT 2020)

# Uncertainty Entangled; Modelling Safety Assurance Cases for Autonomous Systems

Anila Mjeda and Goetz Botterweck

10 pages

Guest Editors: Stylianos Basagiannis, Goetz Botterweck, Anila Mjeda
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

# Uncertainty Entangled; Modelling Safety Assurance Cases for Autonomous Systems

**Anila Mjeda and Goetz Botterweck**

Lero–The Irish Software Research Centre
University of Limerick
Limerick, Ireland
name.surname@lero.ie

**Abstract:** When designing and analysing autonomous systems and their environment it is necessary to consider uncertainty and multiple potential states (of the system and its environment). In this position paper, we explore the idea of notations and modelling tools that are based on *'superpositions'* of states. More concretely, we present a treatment of uncertainty in autonomous systems inspired by quantum physics and propose an extension of the Goal Structuring Notation (GSN), a common approach for the modelling of safety arguments, to model *'superposition'* and *'entangled'* nodes; and, incorporate guidelines of the emerging UL 4600 standard for autonomous systems.

**Keywords:** autonomous vehicles; uncertainty; safety assurance cases; GSN, UL 4600

## 1 Introduction

> I am sitting with a philosopher in the garden; he says again and again 'I know that that's a tree', pointing to a tree that is near us. Someone else arrives and hears this, and I tell him: 'This fellow isn't insane. We are only doing philosophy.
>
> *Ludwig Wittgenstein, On Certainty*

In safety-critical systems, it is critical to give grounds for why the software can be trusted. Current trends on autonomous vehicles (AVs) heighten the challenge of assuring they are safe, especially in view that AVs need to operate in unpredictable situations and to make real-time decisions which so far have been the remit of human drivers. Part of the challenge stands with the fact that developing autonomous systems, entails designing for uncertainty [RJC12, FC19] which in turn exacerbates the need for uncertainty-aware software development methodologies [FC19].

One technique used by classic (not fully autonomous) safety-critical domains is the submission of a *safety assurance case* that provides a documented trail, which demonstrates that state of the art safety considerations have been followed and are traceable throughout the system. This, in essence, entails showing that the potential risks associated with using the safety-critical system in its intended domain have been identified and *acceptably* mitigated. It provides the grounds to convince the certification bodies (e.g. [RTC11, ISO11, FA$^{+}$10, ISO19]) and other stakeholders that the system is acceptably safe [Lev11, HHKM13, DP18].

Using *safety assurance cases* in AVs is still a new and brave frontier. That said, existing research of particular interest for the focus of this paper is the emerging standard[1] for autonomous systems [Und19] and research on treatments of uncertainty [FC19].

In this position paper, we take a multidisciplinary lens to the treatment of uncertainty. We draw ideas from quantum physics where uncertainty has been treated as a domain-native for decades to posit our ideas on modelling uncertainty for autonomous systems. We also build on previous work by [DP18] and [Und19] and propose an extension to the Goal Structuring Notation (GSN)[2].

The remainder of this paper is organised as follows: In Section 2 we discuss *safety assurance cases* and the latest developments in this area. In Section 3 we discuss the concept of uncertainty from a multidisciplinary point of view and present our position in the treatment of uncertainty and its implications for autonomous vehicles (including a running example in Section 3.1). In Section 4 we propose our ideas extending GSN to capture autonomous assurance cases and finally we draw conclusions in Section 5.

## 2 Assurance Cases

Assurance cases are a technique used to argue that a system is safe for a given application in a defined environment. Formally, the assurance case is a *"Reasoned, auditable artifact created for contention that its top-level claim (or a set of claims), is satisfied, including systematic argumentation and its underlying evidence and explicit assumption(s) that support the claim."* [ISO19].

Simply put, assurance cases are structured arguments backed by evidence on why a specific safety goal or claim is met. The typical anatomy of a safety case consists of ***(1)*** *the safety goal (typically split into sub-goals) that needs be achieved;* ***(2)*** *the backing evidence for achieving this goal;* and, ***(3)*** *the structured argument, which establishes the systematic relationship between the evidence and the goals.*

Both the arguments and the evidence can be somewhat informal. As such, the arguments can express some degree of confidence, while the backing evidence can take multiple forms such as results from model checking, testing, analysis, modeling, simulation; or, expert opinions. An in-depth analysis of assurance cases and exiting tool support for them can be found in [DP18]. While there is no particular notation required, the Goal Structuring Notation (GSN) [SCS18], is adopted widely. *Figure 1* provides an illustration of a safety argument in GSN notation constructed for the *Duckietown* environment[3].

The nodes in GSN provide a simple description for the goals, strategies, assumptions,

[1] The Standard for Safety for the Evaluation of Autonomous Products, UL 4600 is being developed as a UL – Underwriters Laboratories standard by a *Standards Technical Panel (STP)* composed from researchers and practitioners from industry and academia. It is projected to be ratified as a UL standard in the first half of 2020.

[2] A graphical notation which is adopted widely to capture safety assurance cases

[3] This simplified diagram is extracted from our work-in-progress research in safety arguments for autonomous vehicles where the Duckietown environment (from an MIT classroom started activity: www.duckietown.org) is used as a running example.

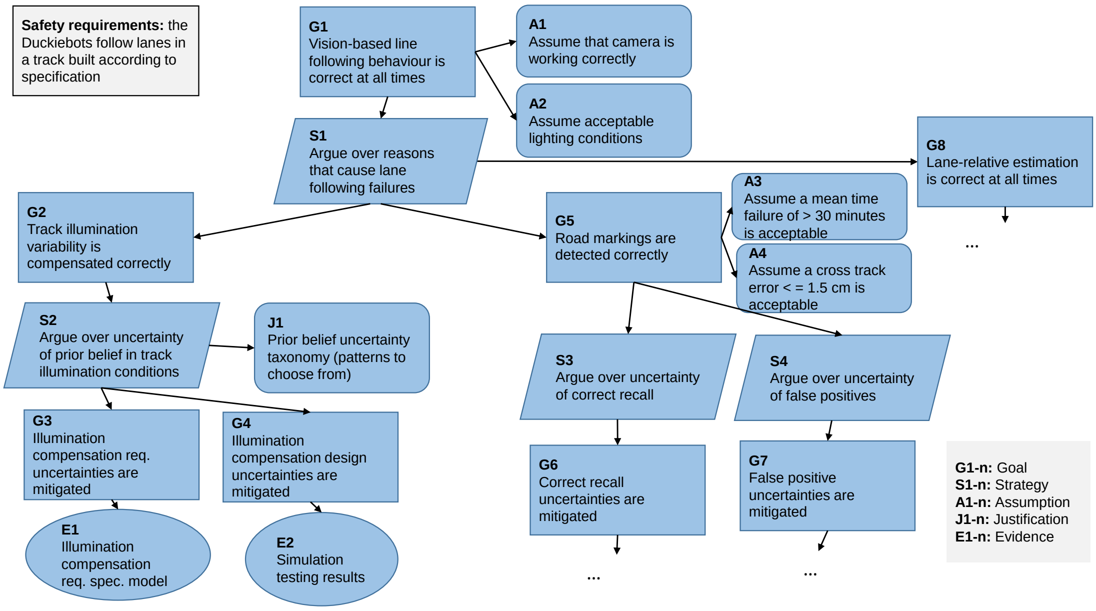
Safety requirements: the Duckiebots follow lanes in a track built according to specification
G1 Vision-based line following behaviour is correct at all times
A1 Assume that camera is working correctly
A2 Assume acceptable lighting conditions
S1 Argue over reasons that cause lane following failures
G8 Lane-relative estimation is correct at all times
...
G2 Track illumination variability is compensated correctly
G5 Road markings are detected correctly
A3 Assume a mean time failure of > 30 minutes is acceptable
A4 Assume a cross track error < = 1.5 cm is acceptable
S2 Argue over uncertainty of prior belief in track illumination conditions
J1 Prior belief uncertainty taxonomy (patterns to choose from)
S3 Argue over uncertainty of correct recall
S4 Argue over uncertainty of false positives
G3 Illumination compensation req. uncertainties are mitigated
G4 Illumination compensation design uncertainties are mitigated
G6 Correct recall uncertainties are mitigated
G7 False positive uncertainties are mitigated
E1 Illumination compensation req. spec. model
E2 Simulation testing results
...
...
G1-n: Goal
S1-n: Strategy
A1-n: Assumption
J1-n: Justification
E1-n: Evidence

Figure 1: A partial safety argument in GSN notation

justifications and evidence and are meant as pointers to fully detailed artifacts. The aim is to trace all of the safety goals to credible evidence without gaps in coverage and where all assumptions are analysed vis-à-vis their validity. Not surprisingly, assurance cases are especially useful in situations where the system is too complex to be ‘proven’ safe using formal verification approaches or more broadly when heavy-weight formal approaches are not feasible to be used.

A promising development for the world of assurance cases is *UL 4600* [Und19] which aims to support a structured way to argue that *autonomous* vehicles are safe. Its first version focuses in *highly automated vehicles* and looks at the challenges of AV safety in the absence of human drivers and presence of non-determinism. It looks at all the different functionality provided by human drivers including fault recovery and expected misuse.

The goal of *UL 4600* is to provide guidance on adopting the required system-level engineering rigour and is meant to be used alongside the other standards that regulate the domain (e.g., ISO 26262) [ISO11]. That said, in contrast to, say, ISO26262, instead of prescribing ‘how to do safety’, UL 4600 is goal-based and makes use of assurance cases to argue in a structured way that the system is *acceptably safe* [PK19]. Typically the safety goal defines what *acceptably safe* means for the case in question and then it proceeds to argue how the AV meets that definition. It does not prescribe any specific software engineering approaches but it requires the use of *rigorous* ones. For example, it does not prescribe how to do testing and verification, but it rather requires a sound argumentation that you have done sufficient testing and verification. Its creators see it as a framework for systematically putting together all of the elements needed to create an *acceptably safe* AV. It facilitates this task by providing reference lists, lists of safety case topics, prompts and epistemic defeaters (‘Did you consider what happens in cases of black ice?’), reference lists of hazards and lists of techniques that can be used to identify hazards, good practices to follow and bad practices to avoid [PK19].

All the prompts provided by UL 4600 need to be considered but not necessarily adopted, and for guidance they are classified into:

- **Mandatory** – prompts that are required to be considered.
- **Highly Recommended** – prompts that can be ignored or altered with non-trivial rationale.
- **Examples** – prompts that are meant as illustrative reminders and not required do be addressed exhaustively.
- **Recommended** –prompts that are entirely optional.

## 3 Reasoning with Uncertainty

Uncertainty has been studied in many disciplines and a significant amount of research has focused on analysing and interpreting it [Hei27, MS58, Laf, TK74, GPS08, TK94, CW04, Kni12].

In economics, typically uncertainty is treated as a stochastic or probabilistic situation [Laf], where the frequency or outcome of an event is not known [GPS08] including situations where it is not possible to specify definite numerical probabilities for a specific (desired) outcome [Kni12]. Treatments of uncertainty in economical studies often intersect with uncertainty in decision making and project management (e.g., [Jaa01, PGW08]).

In psychology, uncertainty is typically discussed against the backdrop of certainty and is conceptualized as a psychological state. Wittgenstein posited that being ‘certain’ is not any ‘knowing’ but it rather is a matter of us feeling that ’we cannot imagine otherwise’[4] [WAV86].

This somewhat non-deterministic treatment of uncertainty resonates with quantum physics where what is commonly referred to as the ’uncertainty principle’ is one of the key concepts that brought forward a revolution on our understanding of the universe.

Uncertainty in quantum physics can be considered ’native’, and yet its meaning is subject to different schools of thought and interpretations. Different conceptualisations of uncertainty include ontological interpretations (lack of knowledge by an observer) and, epistemic interpretations (e.g., measurement inaccuracy). In fact, even the terminology comprises a range of words such as indeterminacy, indefiniteness and, inaccuracy.

At the heart of both the thought revolution; and, lack of consensus and generational debates brought forward by quantum theories is the issue of how (and whether) the measurement process affects the ontological state of an observed system (and in some interpretations the observer her/himself) and it has even entered common parlance as ‘the measurement problem’.

For us, in dealing with uncertainty in autonomous systems, this struggle is mirrored in the somewhat ‘simpler’ struggle of developing a measure of uncertainty that we know how do deal

[4] Please note how Wittgenstein’s thinking connects to Nicolaus Copernicus *(1473–1543)* and to the quantum physicist Hugh Everett III *(1930–1982) who gave us the multiverse quantum hypothesis (referred to as ‘the branching world’ below)*:
“Bryce DeWitt, an American physicist who had edited the journal where Everett’s thesis appeared, wrote a letter to him complaining that the real world obviously didn’t ‘branch’, since we never experience such things. Everett replied with a reference to Copernicus’s similarly daring idea that the Earth moves around the Sun, rather than vice-versa: ‘I can’t resist asking: do you feel the motion of the Earth?’ DeWitt had to admit that was a pretty good response.” – *Sean Carroll, Something Deeply Hidden*

with operationally. Borrowing from the quantum physics' thought experiments we can posit that we can profitably reason about the behaviour of autonomous vehicles if we know a probability distribution of the likelihood that the AV might be behaving in various ways.

*Entanglement* (in quantum physics), at its simplest, suggests that when two particles interact or share spatial proximity, their behaviour becomes intrinsically linked even after they are far apart. Say two electrons were moving with equal opposite velocities and they bump onto each other. Even though before their encounter they each had probabilities of travelling in certain paths which were completely unrelated to each other, after their encounter they will move in *precisely* opposite directions. We say that the electrons are *entangled*[5].

Now let us draw our attention to the 'macro' world of AVs and conceptualise them as entities that have different probability distributions of the likelihood of behaving in various ways. We use the entangled uncertainties metaphor to posit that we should design for uncertainty by conceptualizing our design choices as superimposed ontological representations of reality. Our overall design for uncertainty resembles a 'solutions superposition-model' [6]. Where, each solution taken (in other words the AV's behaviour at a specific time-step matches that particular solution), entangles the uncertainty in the AV behaviour within at least the next time step (the length of the time step would depend on when the next interaction of the AV will cause the uncertainties of its behaviour to become entangled with the environment (e.g., bumps into the next thing). Hence the intuition is that if we have enough 'certain' solutions designed in our 'solutions superposition-model' we can draw upon the 'entanglement of uncertainties' argument and collapse the problem-space into *certain* outcomes for the AV behaviour. In more lay terms, this translates into arguing about narrowing down the all possible behaviours' state-space into a subset of (designed for) acceptably safe behaviour by an AV.

While we characterise possible AV behaviours as probability distributions, another point of frustration we need to clarify is the meaning we give to probability. One school of thought sees the concept of probability connected to the frequency of an event or outcome happening/occurring. That is perfectly satisfying for cases when we can wait to witness the same exact event reoccurring over and over for a large number of repetitions. How often do we get tails if a coin is tossed? But, how do we reason for cases when we are in reality dealing with epistemic probability, where what we already know is important to help us gauge the outcome. Even more so in cases when waiting for a very large number of occurrences or experiments upon some events is not plausible. For example, what is the probability of the soccer team of ones' own country will win the world championship? In these cases, we can reason in terms of attaching beliefs between zero and one hundred percent to the various possible outcomes where the total set of beliefs for the possible outcomes adds up to 100 percent[7].

Next, let us consider some illustrative examples of uncertainties in an autonomous environment.

[5] In the Everettian view of quantum physics the entanglement (e.g., how do the two entangled electrons 'know' what the other is doing) is described via the universal wavefunction, in other words, positing that the quantum state of the whole universe in 'interlinked' and can be 'captured' in one wave function.

[6] Here the superposition-model plays on the quantum concept of superposition

[7] In this regard, there are existing mathematical frameworks such as for example Dempster-Shafer Theory which acknowledge the need for explicitly expressing and quantifying beliefs/confidence/lack of confidence into a certain outcome.

### 3.1 No Duck Left Behind – Uncertainty in *Duckietown*

*Duckietown* [PTA+17] is an environment that started at MIT and is aimed at education and research in autonomous vehicles. *Duckieworld* is comprised of AVs ("Duckiebots") that autonomously drive their users ("Duckies") around their town ("Duckietown") that has roads, traffic lights, and obstacles. The *Duckiebots* can follow lanes while abiding traffic lights and avoiding random obstacles and the *Duckie* pedestrians. Their only sensor is a monocular camera and all their on-board processing is done in a Raspberry Pi.

One of the interesting aspects of the *Duckiebots* for us is that they face a number of software and environment uncertainties (*Figure 1*), that include:

- Prior beliefs on illumination conditions of the roads.
- Prior beliefs on road-markings colours (the *Duckiebots* navigation depends on colour-coded markings).
- Errors in the estimation of fit quality that allows the system to detect road-marking detection failure.
- False positives in road-markings' detection.
- Errors in lane-relative estimation.

In the following section, we use these uncertainties as our running example to illustrate our position in designing for uncertainty.

## 4 GSN Entangled – Designing for Uncertainty

We propose to extend GSN nodes with metadata (building on research by [DP18]) to: a) accommodate our vision of 'solutions superposition-model' to model autonomous systems; and, b) incorporate the extensive reference lists, different types of prompts and epistemic defeaters provided by the emerging UL 4600 standard as natural elements of our treatment of uncertainty ('entangled uncertainties').

Specifically, to account for superposition-model nodes and our treatment of 'entangled uncertainties' we propose to extend the GSN *node types* with:

- nodeType ::= classic | superposition | entangled

Note that when the *node type* is *classic* the node will look exactly as any other classic GSN node. To account for respectively the need to express beliefs in certain outcomes and UL 4600's prompt classifications, we extend the *parameter types* of GSN nodes with:

- belief ::= percentageInDecimalNumber
- promptType ::= mandatory | highyRecommended | example | recommended |

As discussed in the previous sections, these ideas hail from our treatment of uncertainty and more specifically from our thought experiments on 'entangled uncertainties'. In *Figure 2* we illustrate these ideas via a partial assurance case that captures some of the software and environment uncertainties faced by the *Duckiebots* (our running example of an autonomous environment). In here, the overall safety goal that *'Duckiebots will not hit any types of Duckies'*,

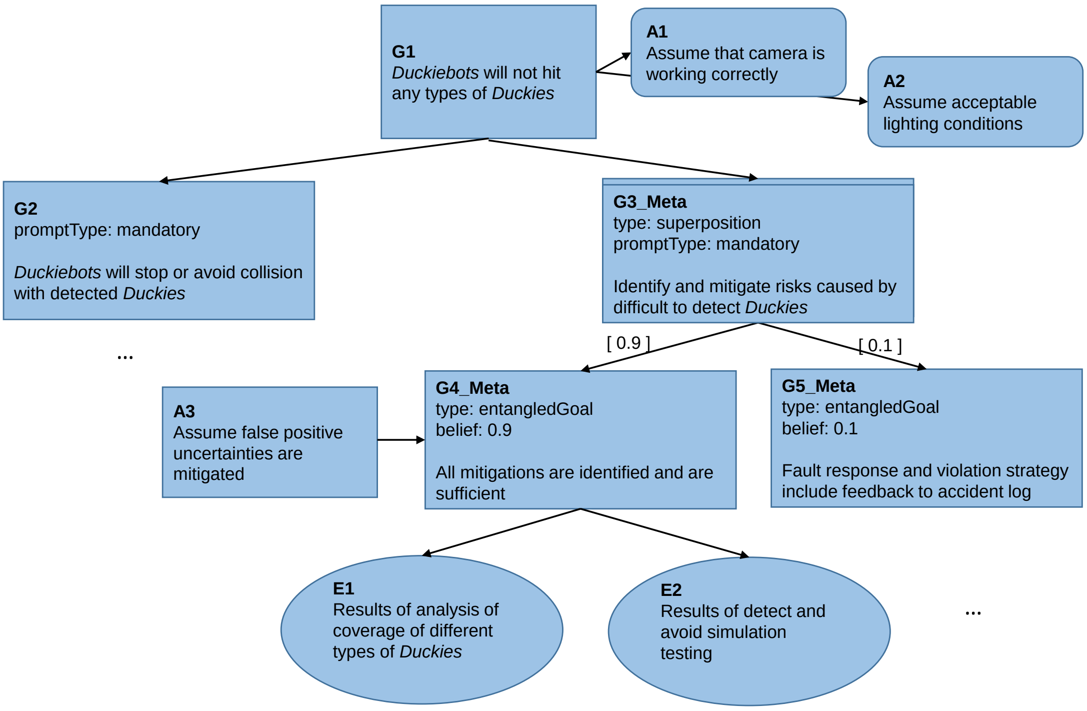
G1
*Duckiebots* will not hit any types of *Duckies*
A1
Assume that camera is working correctly
A2
Assume acceptable lighting conditions
G2
promptType: mandatory
*Duckiebots* will stop or avoid collision with detected *Duckies*
...
G3_Meta
type: superposition
promptType: mandatory
Identify and mitigate risks caused by difficult to detect *Duckies*
[ 0.9 ]
[ 0.1 ]
A3
Assume false positive uncertainties are mitigated
G4_Meta
type: entangledGoal
belief: 0.9
All mitigations are identified and are sufficient
G5_Meta
type: entangledGoal
belief: 0.1
Fault response and violation strategy include feedback to accident log
E1
Results of analysis of coverage of different types of *Duckies*
E2
Results of detect and avoid simulation testing
...

Figure 2: Partial Assurance Case Illustrating Uncertainties in Extended GSN Notation

under the assumptions that *Duckies' cameras are working correctly and they have to operate within acceptable lighting conditions*[8] is split into two sub-goals: 1) *'Duckiebots will stop or avoid collision with detected Duckies'*; and, 2) *'Identify and mitigate risks caused by difficult to detect Duckies'*. The later goal is modelled via a *nodeType: superposition* and both sub-goals have the parameter *promptType: mandatory* in their metadata. The *promptType: mandatory* parameter incorporates UL 4600 in GSN notation, while *nodeType: superposition* reflects our ideas on 'entangled uncertainties'.

We incorporate the concept of 'entangled' probability distributions by modelling the two 'entangled' nodes (*'All mitigations are identified and are sufficient'* and, *'Fault response and violation strategy include feedback to accident log'*) that come out of the superposition node (*'Identify and mitigate risks caused by difficult to detect Duckies'*) by using the parameter type *belief : percentageInDecimalNumber*. This (extended) GSN node' metadata is expressed via a (real) decimal number (or probability), where all the *entangled nodes*' beliefs that come out of a *superposition type node* add up to 1 (100%).

This is particularly important since we believe that as the technology behind quantum computing progresses further we expect to benefit from the real power of notations and modelling tools that 'think' in terms of superpositions $|\psi\rangle$ of the two classical states $|0\rangle$ and $|1\rangle$ coded in quantum memory (for example in the same *qubit*):

[8] Naturally this would translate into accepted weather conditions' in real-world scenarios.

$|\psi\rangle := \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$; where, $|\alpha|^2 + |\beta|^2 = 1$.

In here, the coefficients $\alpha$ and $\beta$ are complex numbers[9] and tie in nicely with our 'entangled nodes' with assigned probability beliefs, since in quantum theories $|\alpha|^2$ and $|\beta|^2$ represent exactly the probability of a state being observed.

Finally, *Figure 2* illustrates examples of evidence (*'Results of analysis of coverage of different types of Duckies'*, and *'Results of detect-and-avoid simulation testing'*) that are used to back up the assurance case's claims.

## 5 Conclusions

The current drive for autonomous systems will benefit from rigorous uncertainty-aware software engineering methodologies. We believe that software engineering could profitably build upon treatments of uncertainty from other disciplines that, in different capacities, have dealt for decades with uncertain environments, measurements, projections, occurrences, etc.

The current safety assurance demands on autonomous systems could benefit from a paradigm shift towards 'complex-state' modelling. This, in conjunction with the gained momentum in quantum computing, could be fostered by notations and modelling tools that 'think' in terms of superpositions and make use of quantum memory (as in *nodes* conceptualised as *quantum states* $|\psi\rangle := \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$; where, $|\alpha|^2 + |\beta|^2 = 1$. and, $|0\rangle$ & $|1\rangle$ represent of the two classical states.)

In this paper, we explore a conceptualisation of uncertainty in autonomous systems inspired by quantum physics which sees uncertainty as a native feature of a system. Furthermore, we propose an extension of GSN (building on research by [DP18]) to: a) accommodate 'superposition' and 'entangled' nodes; and, b) incorporate the extensive reference lists, different types of prompts and epistemic defeaters provided by the emerging UL 4600 standard as natural elements of our treatment of uncertainty ('entangled uncertainties'). In future work, we plan to focus on modelling and validating live feedback that drives the update of beliefs in probability distributions.

## Acknowledgements

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

## Bibliography

[CW04] C. Chapman, S. Ward. Why risk efficiency is a key aspect of best practice projects. *International Journal of Project Management* 22(8):619–632, 2004.

[9] In quantum mechanics they are the amplitudes of the wave function.

[DP18] E. Denney, G. Pai. Tool support for assurance case development. *Automated Software Engineering* 25(3):435–499, 2018.

[FA+10] U. Food, D. Administration et al. Guidance for Industry and FDA Staff-Total Product Life Cycle: Infusion Pump–Premarket Notification. *Draft Guidance (April 2010)*, 2010.

[FC19] M. Famelis, M. Chechik. Managing design-time uncertainty. *Software & Systems Modeling* 18(2):1249–1284, 2019.

[GPS08] I. Gilboa, A. W. Postlewaite, D. Schmeidler. Probability and uncertainty in economic modeling. *Journal of Economic Perspectives* 22(3):173–88, 2008.

[Hei27] W. Heisenberg. Uber den anschaulichen Inhalt der quanten theoretischen Kinematik und Mechanik, Zeit. für Phys., 43, 172–198. *Available in English translation: goo.gl/FMq2J7*, 1927.

[HHKM13] R. Hawkins, I. Habli, T. Kelly, J. McDermid. Assurance cases and prescriptive software safety certification: A comparative study. *Safety science* 59:55–71, 2013.

[ISO11] ISO. ISO 26262: Road vehicles-Functional safety. *International Standard ISO/FDIS* 26262, 2011.

[ISO19] ISO/IEC/IEEE 15026: Systems and software engineering — Systems and software assurance . Standard, ISO/IEC/IEEE International Organization for Standardization, 2019.

[Jaa01] A. Jaafari. Management of risks, uncertainties and opportunities on projects: time for a fundamental shift. *International journal of project management* 19(2):89–101, 2001.

[Kni12] F. H. Knight. *Risk, uncertainty and profit*. Courier Corporation, 2012.

[Laf] J. Laffont. J., 1989, The Economics of Uncertainty and Information.

[Lev11] N. G. Leveson. The use of safety cases in certification and regulation. 2011.

[MS58] J. G. March, H. A. Simon. *Organizations.* Wiley, 1958.

[PGW08] O. Perminova, M. Gustafsson, K. Wikström. Defining uncertainty in projects–a new perspective. *International journal of project management* 26(1):73–79, 2008.

[PK19] D. Prince, P. Koopman. UL 4600 Technical Overview. Technical Presentation at Carnegie Mellon University, Oct 2019.
https://users.ece.cmu.edu/~koopman/talks/191010_UL4600_TECH_Webinar.pdf

[PTA+17] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang et al. Duckietown: an open, inexpensive and flexible platform for autonomy education and research. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Pp. 1497–1504. 2017.

[RJC12] A. J. Ramirez, A. C. Jensen, B. H. Cheng. A taxonomy of uncertainty for dynamically adaptive systems. In *2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. Pp. 99–108. 2012.

[RTC11] RTCA. *DO-178C, Software Considerations in Airborne Systems and Equipment Certification*. RTCA, 2011.

[SCS18] Goal Structuring Notation Community Standard Version 2. Standard, 2018.

[TK74] A. Tversky, D. Kahneman. Heuristics and biases: Judgement under uncertainty. *Science* 185(4157):1124–30, 1974.

[TK94] B. N. Taylor, C. E. Kuyatt. Guidelines for evaluating and expressing the uncertainty of NIST measurement results. 1994.

[Und19] Underwriters Laboratories. Standard for Safety for the Evaluation of Autonomous Products, UL 4600 (Draft). December 2019. https://edge-case-research.com/wp-content/uploads/2019/12/191213_UL4600_VotingVersion.pdf

[WAV86] L. Wittgenstein, G. Anscombe, G. Von Wright. *On Certainty/Über Gewissheit*. Harper Collins, 1986.