Defining an Effective Context for the Safe Operation of Autonomous Systems

Matt Osborne
AAIP, Department of Computer Science
University of York
York, England
matthew.osborne@york.ac.uk

Richard Hawkins
AAIP, Department of Computer Science
University of York
York, England
richard.hawkins@york.ac.uk

***Abstract*—The safety of a system can only be demonstrated to have been achieved in a defined context. This is true whether it is a 'traditional' or autonomous system (AS). For traditional systems, a human is trusted to provide an oversight of operations, and react safely to unexpected scenarios that occur. For AS we cannot necessarily rely on human oversight to handle unexpected events, and must therefore be more confident that all possible hazardous scenarios are understood prior to operation. This makes the task of defining the context of safe operation (CSO) precisely and completely even more important for an AS so that unexpected scenarios can be limited. Attempting to define the CSO completely for an AS operating in a complex open-world environment could be an intractable task. It is therefore imperative that an effective and efficient way to define the CSO for AS can be found.**

Existing approaches to defining the CSO for AS are generally seen to be disjoint (in that each of the elements is considered and specified in isolation) and lacking in focus (in that the level of detail is found to be inconsistent and often inappropriate). What is required therefore is a targeted, iterative and integrated approach for defining the CSO for an AS. We provide an example of how this approach can be used to deliver an effective CSO for an autonomous robot.

***Index Terms*—Safety, Autonomous Systems, Safe Operation, Safe Context**

I. Introduction

It is well understood that safety of any system can only be demonstrated within a defined scope of operation. A safety case for a system is therefore presented within an explicitly defined context. This context of safe operation (CSO) must also be defined for Autonomous Systems (AS). For 'traditional' systems, a human is trusted to provide an oversight of operations [1], and react safely to unexpected scenarios that occur. For AS we cannot necessarily rely on human oversight to handle unexpected events, and must therefore be more confident that all possible hazardous scenarios are understood prior to operation. This makes the task of defining the CSO precisely and completely for an AS even more important so that unexpected scenarios can be limited. Attempting to define the CSO completely for an AS operating in a complex open-world environment could be an intractable task. It is therefore imperative that an effective and efficient way to define the CSO for AS can be found.

This work is funded by the Assuring Autonomy International Programme https://www.york.ac.uk/assuring-autonomy.

Defining the CSO for an AS requires an explicit description of what the AS is required to do, where it is required to do it, and how it should do it. This leads us to consider that the CSO for an AS must consider three distinct yet tightly coupled elements:

1) **The Operational Domain Model (ODM):** the elements with which the AS may be required to safely interact with whilst undertaking its tasks
2) **Autonomous Capabilities (AC):** the capabilities which the system is able to undertake in an autonomous manner.
3) **Operating Scenarios (OS):** the particular set of actions and events that the AS may undertake in order to achieve its objectives [2].

It is vital that all three of these elements are considered when defining the CSO for an AS since they all play a crucial role in determining the safe behaviour of the AS. It is only through understanding what the autonomy can do and where, when, and how that must be done (defined by the AC, ODM and OS respectively) that the hazards and hence the safety requirements for the AS can be determined. As such, it is the interaction between these elements of the CSO that should be the focus of AS safety analysis (as illustrated in Figure 1).

By way of an example, let us consider an autonomous passenger shuttle designed to ferry passengers between an airport terminal and car parks. If a safety analyst wishes to understand the safety impact of the failure of the shuttle's front-mounted LiDAR sensor, this can only be done with full knowledge of what the autonomous shuttle is going to do, the elements of the operating environment it will interact with in doing that (such as road infrastructure, other vehicles, people and weather conditions), and the scenarios that may arise as a result (negotiating junctions with other vehicles or negotiating pedestrian crossings). An analyst must also consider how the safe operation of the shuttle may be affected by changes in elements of the CSO during operation, such as the onset of heavy rain or wind. Understanding the inter-relationship between the different elements of the CSO is crucial. For example, decisions made regarding the scope of the ODM could affect the relevant operating scenarios. This may require trade-offs in terms of tasks performed and the scope of the

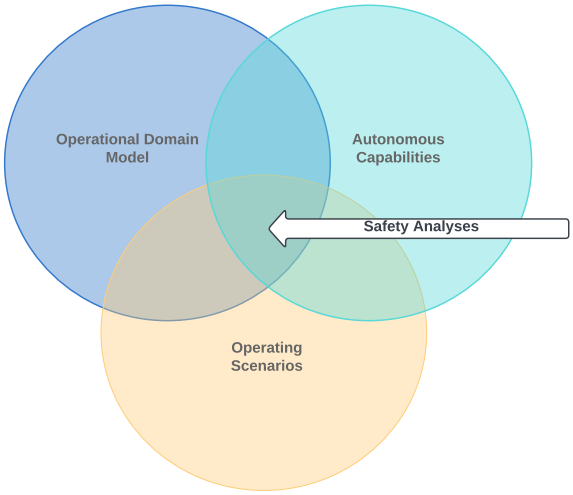

Fig. 1. The Interacting Elements of the CSO

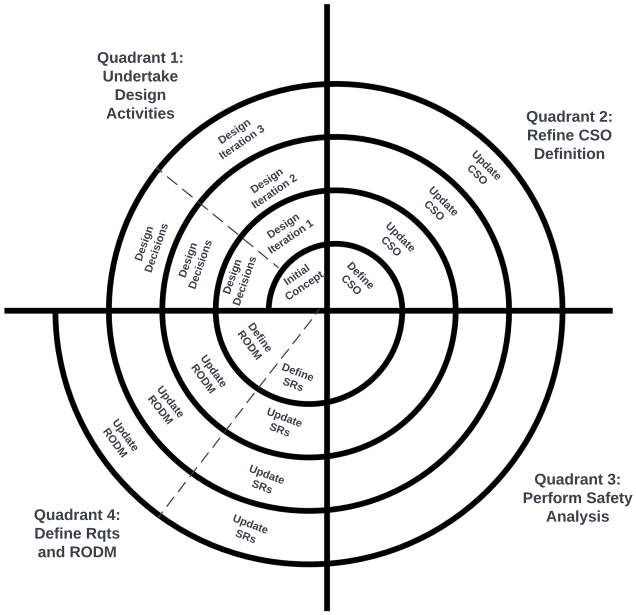

Fig. 2. The Iterative Approach to Defining the CSO

operation which must be justified.

Ensuring the CSO is defined with the correct level of detail is crucial. It must be ensured that all the required information is available and that the specification is neither too sparse nor too full. If the level of detail is too sparse then we cannot be confident in any assertions of safe operation as we may not have identified all foreseeable interactions. Conversely if we require an analyst to consider every conceivable entity in a specific domain of operation we could end up with a state explosion of information without any clear understanding of each entity's importance or relevance. What is required therefore is a targeted approach that enables the CSO to be defined in a coherent manner that prevents a state explosion, and allows the analyst to identify and focus on the aspects most important to the safe operation of the AS (and not on the aspects which don't).

Existing approaches to defining the CSO for AS are generally seen to be disjoint (in that each of the elements is considered and specified in isolation) and lacking in focus (in that the level of detail is found to be inconsistent and often inappropriate). To ensure that the CSO for AS are defined in an effective manner, in this paper we therefore present an iterative approach which results in a more integrated and targeted CSO definition. We present this approach for review and posit it can deliver a CSO which can be justified to be sufficiently complete and correct.

The rest of the paper is structured as follows. We introduce our approach to a CSO for AS in Section II, and consider how the AS can continue to operate safely in the presence of faults, failures, or adverse conditions in Section III. We make concluding remarks in Section IV.

II. An Integrated, Iterative and Targeted Approach

Our approach, as illustrated in Figure 2 (inspired by [3]), requires that the CSO is developed in an iterative manner. This helps to prevent a state explosion of information by ensuring that detail is added to the CSO only when its necessity is identified. As the spiral in Figure 2 traverses the quadrants, the approach follows a cycle of design activities; updates to the CSO; safety analyses (at the intersection shown in Figure 1); and updates to the safety requirements. The final quadrant terminates by placing restrictions on the ODM in the form of a Reduced ODM (RODM). We return to the concept of a RODM once we have discussed the ODM itself.

In this iterative approach, design activities (such as selection of sensor technology, or architectural decisions taken) are the trigger to continue the spiral, and traverse the four quadrants again. A change to any element of the CSO will require a re-assessment of all other elements, and by continuously reassessing, and updating each element of the CSO throughout system development, the approach ensures a tight coupling between the different elements of the CSO to ensure it is developed in an integrated manner which increases in maturity with each design iteration.

Whilst we present the steps to iterate the CSO as the design matures, we are cognisant of the need to integrate the design lifecycle with the CSO. We aim to formalise this link as we mature the design and the CSO of an autonomous robot we are developing (see Section II-E).

We next discuss the construction of each of the three elements of the CSO in turn, before going on to discuss how the CSO is elicited, developed, and matured.

A. The Operational Domain Model

In this paper we focus primarily on the construct of an ODM. We do this for two reasons. Firstly, if not carefully managed, the ODM could manifest in a state explosion of

data - the majority of which would have no material impact on safety. Secondly, the ODM represents the logical model on which the relationships of all elements of the CSO can be assessed as a point of reference.

The ODM must represent the variables in the environment with which the AS must safely interact with at run-time.

The current means by which the operational environment is defined and represented uses the example from the automotive industry which has started to consider how the operation of Autonomous Vehicles (AV) (or automated systems therein) can be safely assured through the use of ODDs, but there are many definitions of what constitutes an ODD for AVs. These include:

- The specific conditions under which a given driving automation system or feature thereof is designed to function, including, but not limited to, driving modes [4]
- Everything that an automotive system can be exposed to on the road [5]
- A subset of all the possible situations that could be dealt with by a human driver [6]
- A set of driving conditions under which a certain Automatic Driving System (ADS) is designed to function [7]
- An abstraction of the operational context for an ADS [8]
- The domain over which an automated vehicle can operate safely [9].

Khastgir notes that everyone has their own understanding of what an ODD is, (either) to fit their products or preconceived notions [10], and Heyn et al found there was no common definition for an ODD [8]. Whilst these differences in interpretion exist, it is hard to conceive how ODDs can be trusted as a mechanism for achieving and demonstrating AS safety.

Czarnecki [11] asserts that an ODD may place limitations on the environment, AS behaviours, and the state of the vehicle. **We argue this is the role of safety requirements**. Czarnecki also argues that the ODD may reflect the requirements of a particular driving automation feature. **We argue that an ODD should not contain requirements' traces**.

Gyllenhammar et al [5] take the work of [6] and [11] to argue that an ODD is used to model and collect the operating conditions for an AV; with the ODD's primary purpose being to confine the safety analyses to only what is necessary (as long as the models are complete, correct and sufficient). As we will argue, this should be the other way round - the 'models' need only be as granular as required, but the safety analyses must always be complete, correct, and defensible.

Khastgir et al introduced the notion of 'Informed Safety' for AVs [12] where drivers are "informed about the safety limits of the automated system to enable them to calibrate their trust in the system to an appropriate level". In a more recent article, Khastgir enhances the notion of Informed Safety to include the aspect of enabling a user to be aware of what a system can and cannot do (by understanding the conditions in which an AV is capable of operating safely) [10]. Khastgir defines an ODD as being the operating environment for which a system is designed for, and able to operate safely within (ibid).

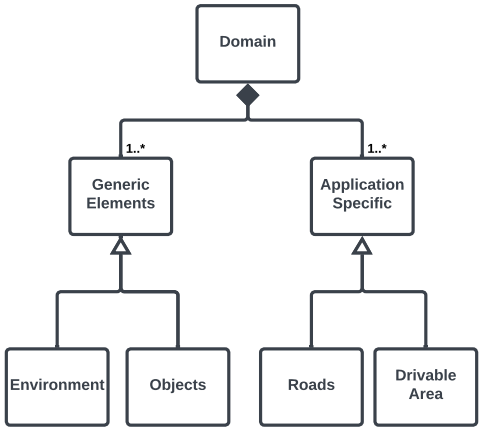

Fig. 3. Domain Elements of an ODM

We are aligned with Khastgir's definition of what constitutes an ODD, but because of the confusions and differing assertions that exist regarding the contents, and role of an ODD, we don't believe ODDs are sufficient for the CSO for AS. We have therefore established a simple, alternative, and distinct model which *builds on* the concept of an ODD, and which we propose for use with all types of AS, and for all levels of autonomy. This alternative model we refer to as the Operational Domain Model (ODM). The purpose of the ODM is to inform safety analyses. For clarity, we break down the ODM into its three constituent parts:

- **Operational:** The operating parameters (e.g. time and duration of operation) and any restrictions on AS operation
- **Domain:** The environment in which the AS will operate
- **Model:** The pictorial representation of the ODM.

1) Creating an ODM: An ODM is constituted by the Operational and Domain elements, which are now considered in turn. Figure 3 shows the construction of the domain elements of an ODM, which is constituted by generic elements (i.e. the meteorological environment and expected objects) and application-specific elements such as roads and drivable areas for an AV.

In the ODM Figures, the white, triangular arrow heads denote an 'instance of' (e.g. environment is an instance of generic elements); the black, diamond arrow head signifies 'constitutes' (e.g. the domain is constituted by the generic and application specific elements); and the numbers on the arrows denote multiplicities (e.g. at least the number of the first given figure).

The operational aspects of the AS are kept as simple and as precise as possible, although we would always expect to see the 'Time Metric' (when the AS may operate), and a 'Duration Metric' (the permitted/required duration of operation) as shown in Figure 4.

We may also expect to see elements such as 'Restrictions' (i.e. prohibited zones) placed on the AS in this part of the ODM. Such restrictions are not safety requirements per se (although they may unintentionally contribute to safety), but

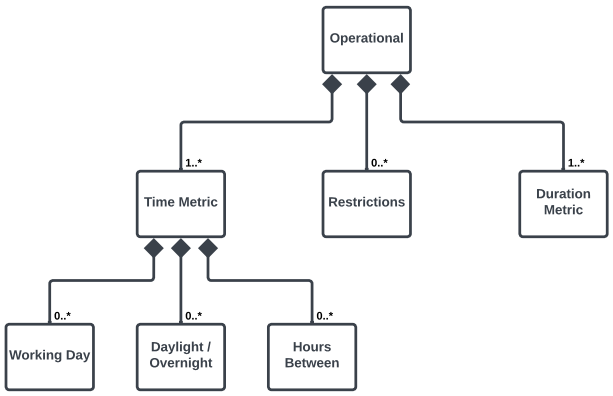

Fig. 4. Example Operational Element of an ODM

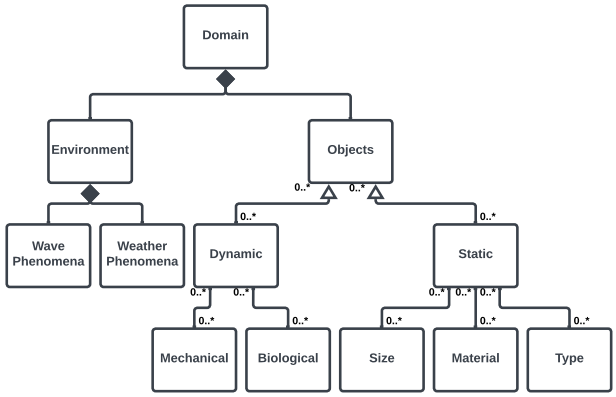

Fig. 5. Generic Domain Element

are imposed on the AS by the customer. Ordinarily, information regarding time, duration, and any restrictions will emanate from User Requirements in the first instance.

We identified two potential, yet opposite approaches for establishing the Domain element. Option 1 is not preferred as it requires the analyst to identify and classify every conceivable variable in the environment which could ever manifest. The second approach is a more pragmatic and balanced approach, and requires the analyst to provide only the minimum amount of detail **in the first instance**. To illustrate the contrast between the two approaches, we consider each option in turn.

Option 1 requires that **every conceivable object that may be present in the operating environment of the AS** is placed in a robust taxonomy which decomposes into entities, instances, dimensions, textures, and colours etc. The logical consequence of this approach is requirements for the AS to detect and classify each of these entities. This has implications on the selection of sensor technology, and for the data required to train and validate detection and classification algorithms. Such an approach presents two significant developmental risks - over-engineering the design solution (should specific capabilities not, in fact be required), and over-fitting the algorithmic training data [13].

As an example consider potential dynamic objects in the environment. Under Option 1, the analyst would need to further decompose these objects into biological and mechanical instances and then further by considering specific features, which for biological instances may include:

- Classification (animal/human)
- Dimensions (height, body type, age)
- Clothing (colours or, and any headgear)

When considering static objects such as permanent fixtures, the analyst would need to further decompose them by considering aspects including (but not limited to):

- Type of fixture (e.g. office furniture, or white goods)
- Dimensions
- Uniformity of dimensions (i.e. is it a perfectly rectangular cabinet, or multi-castored office chair?)
- Colour
- Texture
- Opacity (i.e. a translucent, or an opaque door/wall).

Much of this data may not have a material impact on the safe design of the AS, nor its ability to safely operate, and following such an approach may not be commensurate with risk (as we simply do not know enough about the contribution to hazards at the early design phases).

Option 2 provides a more pragmatic approach insofar as the first draft of the 'Domain' element **contains only a high-level description of the dynamic and static objects** that are expected to be encountered by the AS. For example, for an office environment, this requires only an initial high-level overview:

- **Dynamic Objects** (biological / mechanical)
- **Static Objects** (furniture / building features such as walls, floors, or doors)

For the initial representation of the **Domain** it is not (yet) possible to assert whether the AS should be able to classify objects - only that they should be detected (to maintain a safe separation). We can only identify any requirement to classify objects through analysis of the entire CSO.

It is only once we know the capability of the AS and its required operating tasks that we need to consider a more detailed model of the **Domain**, and any need to update the model will be identified through safety analysis performed against the CSO, as we will discuss later.

We posit that **Option 2** presents the more pragmatic and efficient route to establishing and managing the **Domain**, as the required detail is identified and matured commensurate with the identified risk, and is commensurate with the objective of **only modelling the variables of interest**.

In Figure 5, we illustrate an example of a generic domain model. For the environment, we expect to see wave phenomena (aspects of the Electromagnetic Spectrum) and weather phenomena. Only the types of phenomena are considered at the initial stage, as their existence may not have a material impact on safe operation (and if it does, we don't yet know

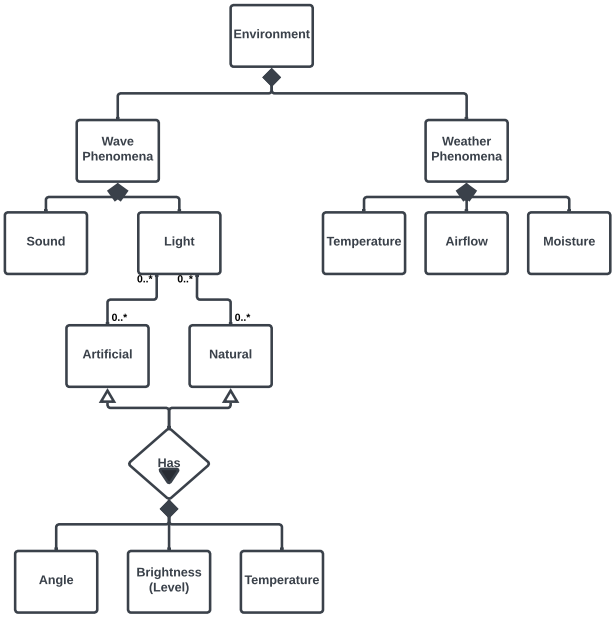

Fig. 6. Generic Environment Element

how). Examples of how the Wave and Weather Phenomena may be decomposed are shown in Figure 6.

As an example, consider the use of the entity 'Moisture'. The domain model could include more detail on this aspect such as 'rainfall'. However it may not be the manner by which moisture appears (rainfall, spillages) that matters - more that the presence of surface water may appear on the drivable area. Alternatively, moisture could result in a reduction in visibility of a visual sensor. Both instances are a potential safety concern. Whether we need to detect rainfall, or moisture on drivable surfaces, and/or or the obscuring of lenses may not be known until the required tasks of the AS are understood, and design decisions are taken (such as the type of camera selected, or the traction of the tyres is known). Therefore more detail on this entity should not be added to the domain model unless and until such information is known.

We illustrate the 'Objects' aspect of a generic ODM in Figure 5 (dynamic and/or static objects expected to be present. We further decompose Dynamic Objects into mechanical (which may or may not be capable of movement under their own volition), and biological (humans and/or animals). We decompose Static Objects by considering their size, material, and type.

It is not yet necessary to model all potential variables of both dynamic and static objects (until further safety analyses are undertaken), it *may* be sufficient for us to consider size ranges alone initially. As design decisions are taken, the required granularity of detail in the model will be revealed further.

B. Reduced Operational Domain Model (RODM)

Although AS are designed to operate 'within the ODM' it is inevitable that the AS will move 'outside' of the ODM as system and/or environmental variables manifest. The manifestation of these variables may make the operation of the AS unsafe immediately (in which case the AS may need to hand over to human control, or perform a minimum risk manoeuvre) - or may necessitate the application of restrictions on the operation of the AS.

Here we build on the work of Colwell et al [9] and their proposal to establish a *Restricted* Operational Domain (ROD). The theory behind creating a ROD was to specify a domain within which a degraded automated driving system is still able to function safely (albeit at a reduced functionality).

A potential limitation of Colwell et al's use of a ROD is that it is limited to consider only how changes in system variables may place an AS in a ROD, whereas we have enhanced their work to also include consideration of environmental variables (both in isolation AND in conjunction with system variables). We refer to this concept as the *Reduced* Operational Domain *Model* (RODM).

As an AS design matures, and the elements of the CSO are assessed for potential impact on the safe operation of the AS (i.e. conducting a safety analysis of the CSO), system and environmental triggers which would place the AS as operating outside of the ODM are identified. Consider our robot which is required to transport small packages from A to B, and operate in all light levels. Safety analysis of the operating scenarios, autonomous capabilities, and system and environmental variables reveals combinations of variables which would trigger entry to an RODM, or make the operation unsafe. One aspect of these system variables relates to the AC of the AS.

C. Autonomous Capabilities

The AC describe the functions and tasks which the system can undertake autonomously. The automotive industry uses Levels of Autonomy [4] to define the overarching capabilities of vehicle types, but these are not a sufficiently granular means of defining the specific capabilities of an AS.

For defensible and compelling safety analyses to be undertaken, it is vital that the AC of the AS are accurately defined, and updated as the design is matured. For the initial draft of the AC, the analyst can only hope to define the required capabilities of the AS at a very high level of abstraction, but the level of detail will increase as the design matures, and trade-offs such as cost / benefit are decided on.

In their 2018 paper on informed safety, Khastgir et al [12] argue that knowledge of the AC *and* the known limitations should be stated in order to develop trust in the system (which they define as a "history dependent attribute that an agent will help achieve an individual's goals in a situation characterised by uncertainty and vulnerability"). In the paper, they propose three levels of knowledge with which trust can be built, and we argue that two of these can be used as a structure of knowledge required for construction of the AC model:

- **Static Knowledge:** an understanding of the functionality of the AS. Whilst a driver/user of an AS could build on static knowledge over time (usage) we restrict the use of static knowledge to purely what is known from a design point of view. As the design matures, this should also include limitations (e.g. should there be a cost trade-off for a low-grade camera [12], safety analyses will be better informed with an understanding of any limitations in perception during periods of reduced visibility).
- **Real Time Knowledge:** or dynamic knowledge about the automated system(s). Here we use the types of data that are intended to be fed to an operator / monitor (either human / machine / both) (and how), in order to better inform safety analyses. This should also include the means by which such data is provided.
- **Internal Mental Model:** prior beliefs influenced by external sources such as marketing material. Whilst this *could* be used to establish the trust in the system, it adds little value for informing the creation of an AC model.

1) Creating the Autonomous Capabilities: Restricting our efforts to represent Static and Real Time Knowledge, we can define the AC. In the initial AC model one might only instantiate the basic AC predicated on a Sense-Understand-Decide-Act model which states how the AS:

1) **Senses:**
   - perform object identification
   - identify environmental variables
2) **Understands:**
   - classify objects
   - perform localisation
3) **Decides:**
   - conduct mission planning
   - motion planning
4) **Acts:**
   - realise changes in speed and direction

D. Operating Scenarios

The current means to represent the OS differ between industries (i.e. the Concept of Operations (CONOPS) in aerospace and Use Cases/Operating Scenarios in the automotive industry, and we find no fault with these approaches. Our approach *builds upon* these approaches by introducing a graphical representation.

1) Creating the Operating Scenarios: OS are elicited as a specific instance of scenarios [14]. As OS are also conditional on the elements of an ODM, it can become increasingly complex to maintain the maturing OS in textual form - as can be seen for a single OS for an AS following a planned path in an office environment:

- AS is following planned path, and
  - Sender is in an inaccessible location, and/or
  - Lighting within building becomes too dark/bright, and/or
  - Doors are present between robot and destination, and/or
  - Prohibited doors (transparent/revolving/not controllable remotely) are present between AS and destination, and/or
  - A static object is in the path/trajectory of the AS, and/or
  - A dynamic object is in the path/trajectory of the AS, and/or
  - A forbidden zone is on the route of the planned path, and/or
  - A blind corner exists on the AS' planned path, and/or
  - AS runs out of charge, and/or
  - AS develops a fault, and/or
  - AS fails to arrive at destination, and/or
  - Robot becomes trapped between objects.

As such, we have found Activity Diagrams (such as those described in [1]) a useful tool to manage this complexity.

E. Integrating the Different Elements of the CSO

In this section we illustrate the integration and development of the CSO by reference to autonomous robots being developed at our University. The use of bold font denotes the quadrant number shown in Figure 2 as the process iterates.

The initial concept design (**quadrant 1**) specified the design intent for robots capable of delivering small packages around an office. The objective was for occupants to request a robot to come to them anywhere in the building and deliver a package to a desired destination.

As we moved to **quadrant 2**, we defined the initial version of the CSO. In the initial version of the AC Model, only a conceptual design existed, so we restricted our considerations to the generic autonomous functions of Sense, Understand, Decide, and Act, with processing distributed between these functions.

The initial OS was defined entirely based on the robot's Use Cases, Exception Cases, and Preconditions. For the ODM, the only relevant information for the operational element available to us related to restrictions (stairways and WCs would be prohibited areas), and duration (the robots would only be required for some duration during a typical working day). The domain element was initially limited to consider the variables that would constitute the office environment (zones, comprising rooms, corridors, stairs, and a lift; with rooms comprising walls, floors, and doors).

A Decision-Safety Analysis [1] was carried out in **quadrant 3** followed by a more detailed, yet targeted HAZOP analysis. The outputs from these analyses were used in **quadrant 4** to define an initial set of safety requirements predicated on maintaining a safe separation minima. The results of the HAZOP identified additional requirements on the detection of dynamic objects within a predetermined distance, and the ability to detect specific static objects which may present difficulties to available sensor technologies.

It was only possible to elicit meaningful safety requirements through the consideration of the CSO as a whole (i.e. the relationships between technological capability (AC), the objects in the environment and environmental conditions (ODM), and

the tasks being performed by the AS at the specific instance in time (OS).

The safety analyses performed against all elements of the CSO also identified the system and environment variables that would trigger an exit from the ODM.

In making design decisions (selecting the sensors, operating system, and detection algorithms), we move to **quadrant 1** of another iteration of the spiral in Figure 2. We are now able to update in **quadrant 2** the AC with the specifics of *how* the robot will sense, understand, decide, and act. We also updated the ODM to include the values for the light angles, brightness and temperature (which we had established through the detailed building design); along with the specific details of office furniture we had identified in quadrant 4 that it was revealed presented difficulties to the selected sensor technology in certain environmental conditions (a cardboard box, a 5-castor office chair, and a transparent container in this intance).

The selected sensor technologies were then tested for their ability to detect the static objects in the ODM (specifically those objects deemed to be 'difficult' to detect) as part of the safety analyses (**quadrant 3**). This was carried out under laboratory conditions for different variations of light-levels (including darkness), -temperatures, and -angles. In summary, we found that no single sensor was capable of detecting all three object types under all lighting conditions, and that the Point Cloud and the LiDAR represent the Minimum Equipment List (MEL) for static object detection when lighting is unavailable. This both derived safety requirements for the MEL (**quadrant 4**), and also identified the combinations of system and environmental variables which would place the robot outside the ODM should they manifest.

The identification of the system and environmental variables which would place the robot in a RODM then informed the need for a design decision for the acquisition of health and status monitoring solutions - and the spiral continued once more in to **quadrant 1**.

III. Monitoring the ODM

The ODM represents those variables that the AS will encounter and must be capable of safe interaction with, and whilst the system and environmental variables therein modelled are 'true' the AS is held to be operating 'inside the ODM' (and therefore operating safely). There will be occasions when environmental and/or system variables occur which place the AS 'outside of the ODM'. Excursions from the ODM may not necessarily be unsafe immediately, and it may be that the AS can continue to operate safely, albeit with some form of diminished capability.

A. Monitoring Domain Models

Colwell et al [9] asserted the need to monitor ODDs and RODs and we are fully aligned with their approach. To assure the continued safe operation of an AS, we need to know when:

- The AS is operating inside the ODM
- The AS is operating inside the RODM

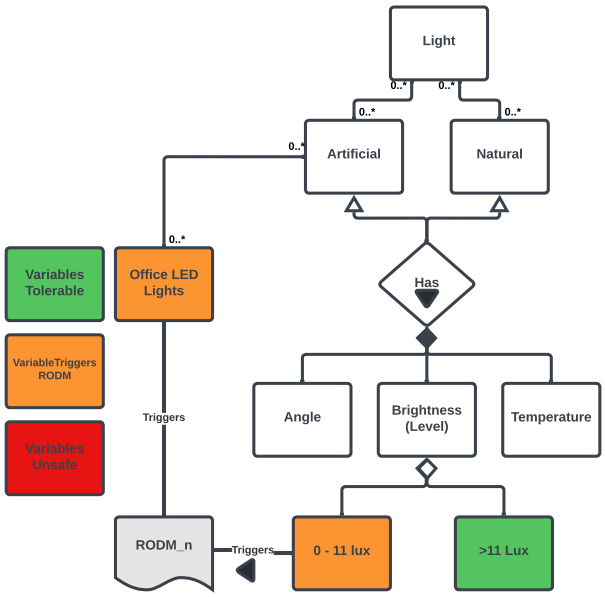

Fig. 7. Domain Variables of Interest

- The AS is operating unsafely.

To assure continued safe operation, we therefore need to identify the system and environmental variables which - either singularly or in combination - would place the AS outside of the ODM. This requires us to monitor identified variables of interest. Similarly, we also need to monitor for which values of variables will place the AS *back inside* the ODM.

The first step to elicit monitoring requirements is to identify the variables of interest to monitor. Key here is to identify the precise variables of interest. For example, whilst rainfall may be a causal factor in obscured sensors, or slippery surface conditions, it is the obscuring of sensors, or loss of friction which are the variables of interest for monitoring purposes, and not necessarily the causes thereof. Whilst readily-available, and reliable technology exists for detecting whether a sensor is obscured, and reliable technology also exists for monitoring friction, the same cannot yet be said for meteorological forecasting at exact location co-ordinates (as it may be already too late once precipitation has commenced). As such not only does identifying specific variables of interest prevent a state explosion (of monitoring) it may also lead to a simpler and more reliable solution to instantiate into a design.

So how do we systematically identify the system and/or environmental variables which require monitoring? Herein lies a benefit of having a understanding of all elements of the CSO. A systematic, deviation-based safety analysis of the CSO readily identifies whether a variable could detriment safety, and under which conditions.

Treating each modelled element as a logical node, deviation-based analysis can be applied to the models as shown in Figure 7. Here it can be seen that a light level of 11 Lux

or under, in conjunction with a lighting failure could lead to an unsafe condition should there be a concurrent failure of the Infrared camera sensor; which would render the AS effectively 'blind'. This combination of failures/events may be unsafe when considering the operating scenario of the AS.

Assessment of the CSO elements allows the analyst to identify a set of variables which (either singularly, or in combination) will trigger an RODM.

This manifests in a need for three variables to be monitored:

- Light levels of the operating environment
- Health (serviceability) of the Infrared Camera
- Operational status of the streetlights which illuminate the forward-facing operating area of the AS.

There is also a requirement for some secondary monitoring. As these triggers require the AS to moderate variables which it has direct control over (speed and/or direction), we need to be confident that the required action has indeed occurred (i.e. the AS has slowed down to a new maximum permissible speed for safe operation).

Implicit in this secondary monitoring is the further consideration of what actions must be taken should the AS *not* moderate its speed under such conditions.These same monitors must also be capable of identifying when system and/or environmental variables will place the AS either *back* into the ODM, or perhaps in an *unsafe* condition. Considering the former case, the AS must *reverse* any changes to speed and/or direction once it is again operating inside the ODM. The AS must 'be aware' of any variables which place it under unsafe operations and take immediate action to place it in a safe state. We do not consider in depth the specifics of such actions in this paper, but any actions should place the AS in a state and condition where it presents the lowest risk of harm (e.g. moving to a place of least obstruction).

Any elicited safety requirements which place an AS inside a RODM/declare its operation unsafe *and* which also return an AS to safe operation, must consider levels of required hysteresis in order to prevent a repetitive cycle of moderating speed/direction as variables fluctuate. An example of this would be fast-moving clouds which may result in light-levels which change rapidly to values either side of 11 Lux (thereby initiating a rapid alternating cycle of ODM - RODM - ODM - RODM as light levels alter).

IV. Conclusion

In this paper we have proposed an iterative, integrated, and targeted approach for defining the CSO for an AS. The CSO is constituted by three tightly coupled elements: ODM (and the RODM), AC, and OS. We have described how these elements are developed using a cycle of design activities; updates to the CSO; safety analyses; and refinement of the safety requirements. This spiral of development of the CSO prevents a state explosion of information, and restricts the elements of the CSO to contain only those variables that matter to safety. We have also shown how these variables of interest are used to identify excursions into reduced and unsafe operation. Further iterations of the process are required before we can be confident in the effectiveness of our proposed approach through-life, however.

We are cognisant of the fact that we have only demonstrated our approach in a single application, so will perform further validation through application to other systems. We currently only model the CSO manually, so will look to explore more formal ways of modelling including the consideration of MBSE tools with robust ontologies.

We are also aware that our proposed process for the careful management of a CSO for AS will be dependent on effective integration with safety engineering practices, and acknowledge that this integration does not yet exist.

Current means for eliciting and managing the context of safe operation is not sufficient for the safety of AS. Despite the need for future work, we submit our approach for review and consideration as a potential solution to the shortfalls of current means.

References

[1] M. Osborne, R. Hawkins, and J. McDermid, "Analysing the safety of decision-making in autonomous systems," in *Computer Safety, Reliability, and Security: 41st International Conference, SAFECOMP 2022, Munich, Germany, September 6–9, 2022, Proceedings*, pp. 3–16, Springer, 2022.

[2] R. Hawkins, M. Osborne, M. Parsons, M. Nicholson, J. McDermid, and I. Habli, "Guidance on the safety assurance of autonomous systems in complex environments (sace)," *arXiv preprint arXiv:2208.00853*, 2022.

[3] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, 1988.

[4] SAE, "SAE J3016. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 2018.

[5] M. Gyllenhammar, R. Johansson, F. Warg, D. Chen, H.-M. Heyn, M. Sanfridson, J. Söderberg, A. Thorsén, and S. Ursing, "Towards an operational design domain that supports the safety argumentation of an automated driving system," in *10th European Congress on Embedded Real Time Systems (ERTS 2020)*, 2020.

[6] P. Koopman and F. Fratrik, "How many operational design domains, objects, and events?," in *Safeai@ aaai*, 2019.

[7] N. Reddy, H. Farah, Y. Huang, T. Dekker, and B. Van Arem, "Operational design domain requirements for improved performance of lane assistance systems: A field test study in the netherlands," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, pp. 237–252, 2020.

[8] H.-M. Heyn, P. Subbiash, J. Linder, E. Knauss, and O. Eriksson, "Setting ai in context: A case study on defining the context and operational design domain for automated driving," *arXiv preprint arXiv:2201.11451*, 2022.

[9] I. Colwell, B. Phan, S. Saleem, R. Salay, and K. Czarnecki, "An automated vehicle safety concept based on runtime restriction of the operational design domain," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1910–1917, IEEE, 2018.

[10] S. Khastgir, "The curious case of operational design domain: What it is and is not?," 2020. Accessed 26-05-2022.

[11] K. Czarnecki, "Operational design domain for automated driving systems," *Taxonomy of Basic Terms, Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo, Canada*, 2018.

[12] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, "Calibrating trust through knowledge: Introducing the concept of informed safety for automation in vehicles," *Transportation Research part C: Emerging Technologies*, vol. 96, pp. 290–303, 2018.

[13] R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu, and I. Habli, "Guidance on the assurance of machine learning in autonomous systems (amlas)," *arXiv preprint arXiv:2102.01564*, 2021.

[14] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *2015 IEEE 18th international conference on intelligent transportation systems*, pp. 982–988, IEEE, 2015.