AERoS: Assurance of Emergent Behaviour in Autonomous Robotic Swarms*

Dhaminda B. Abeywickrama†
Department of Computer Science
University of Bristol, UK
dhaminda.abeywickrama@bristol.ac.uk

James Wilson†
Department of Engineering Mathematics
University of Bristol, UK
wilson-james@outlook.com

Suet Lee
Department of Engineering Mathematics
University of Bristol, UK
suet.lee@bristol.ac.uk

Greg Chance
Department of Computer Science
University of Bristol, UK
greg.chance@bristol.ac.uk

Peter D. Winter
School of Sociology
University of Bristol, UK
peter.winter@bristol.ac.uk

Arianna Manzini
Bristol Medical School
University of Bristol, UK
amanzini.ethics@gmail.com

Ibrahim Habli
Department of Computer Science
University of York, UK
ibrahim.habli@york.ac.uk

Shane Windsor
Department of Aerospace Engineering
University of Bristol, UK
shane.windsor@bristol.ac.uk

Sabine Hauert
Department of Engineering Mathematics
University of Bristol, UK
sabine.hauert@bristol.ac.uk

Kerstin Eder
Department of Computer Science
University of Bristol, UK
kerstin.eder@bristol.ac.uk

Abstract

The behaviours of a swarm are not explicitly engineered. Instead, they are an emergent consequence of the interactions of individual agents with each other and their environment. This emergent functionality poses a challenge to *safety assurance*. The main contribution of this paper is a process for the safety assurance of emergent behaviour in autonomous robotic swarms called *AERoS*, following the guidance on the Assurance of Machine Learning for use in Autonomous Systems (AMLAS). We explore our proposed process using a case study centred on a robot swarm operating a public cloakroom.

***K*eywords** Assurance · Safety · Emergent Behaviour · Guidance · Swarms

* †D.B. Abeywickrama and J. Wilson contributed equally.

1 Introduction

Swarm robotics provides an approach to the coordination of large numbers of robots inspired by swarm behaviours in nature [1]. The overall behaviours of a swarm are not explicitly engineered in the system. Instead, they are an emergent consequence of the interactions of individual agents with each other and the environment [2]; this poses a challenge to *assurance*. According to the ISO standard for systems and software engineering vocabulary [3], *assurance* is defined as "all the planned and systematic activities implemented within the quality system, and demonstrated as needed, to provide adequate confidence that an entity will fulfil requirements for quality". Assurance tasks comprise conformance to standards, verification and validation (V&V), and certification. Assurance criteria for autonomous systems (AS) include both functional and non-functional requirements such as safety [4].

Existing standards and regulations of AS are either implicitly or explicitly based on the V lifecycle model [5], which moves from requirements through design onto implementation and testing before deployment [6, 7]. However, this model is unlikely to be suitable for systems with emergent behaviour (EB); for example through interaction with other agents and the environment, as is the case with swarms. ISO standards have been developed for the service robotics sector (non-industrial) (e.g. ISO 13482, ISO 23482-1, ISO 23482-2), and the industrial robotics sector (e.g. ISO 10218-1, ISO 10218-2, ISO/TS 15066) [2]. However, although these standards focus on ensuring the assurance of robots at the individual level, they do not cover safety or any other extra-functional property at the *swarm* level that may arise through EB.

The main contribution of this paper is a process for the safety assurance of EB in autonomous robotic swarms (AERoS), adapted from the guidance on the Assurance of Machine Learning for use in Autonomous Systems (AMLAS) [8]. AERoS covers six EB lifecycle stages: safety assurance scoping, safety requirements elicitation, data management, model EB, model verification, and model deployment. The AERoS process is domain independent and therefore can be applied to any swarm type (e.g. grounded, airborne). In this paper, we explore it using a case study centered on a robot swarm operating a public cloakroom at events with 50 to 10000 attendees [9]. In the cloakroom, a swarm of robots assist attendees to deposit, store, and deliver their belongings (e.g. jackets) [9]. As the swarm operates in a public setting, the system must prioritise public safety.

The rest of the paper is organised as follows. In Section 2, we provide key related work to our study. Section 3 discusses the six stages of the AERoS process. Finally, Section 4 provides a brief discussion and concludes the paper.

2 Related Work

AS are considered to follow a much more iterative life-cycle compared to the conventional V-model. Thus, there is a need for new standards and assurance processes that extend beyond design time and allow continuous certification at runtime [10]. In this context, there have been several standards and guidance introduced by various industry committees and research groups. In 2016, the British Standards Institution introduced the BS 8611 standard that provides a guide to the ethical design and application of robots and robotic systems. Then, IEEE through its Global Initiative on Ethics of Autonomous and Intelligent Systems initiated the development of a series of standards to address autonomy, ethical issues, transparency, data privacy and trustworthiness (e.g. IEEE P7001 [11], P7007, P7010). There are several standards and guidance related to machine learning in aeronautics, automotive, railway and industrial domains [12], for example the AMLAS process [8], the European Union Aviation Safety Agency (EASA) concept paper, the DEpendable and Explainable Learning (DEEL) white paper, the Aerospace Vehicle System Institute (AVSI) report, the Laboratoire National de Métrologie et d'Essais (LNE) certification, and the UL 4600 standard. However, none of these approaches targets robot swarms.

In this work we used AMLAS [8] as the foundation for developing an assurance process for autonomous robotic swarms. AMLAS provides guidance on how to systematically integrate safety assurance into the development of the machine learning components based on offline supervised learning. AMLAS contains six stages where assurance activities are performed in parallel to the development activities. AMLAS has the advantage of ensuring safety assurance for complex AS where the behaviour of the system is controlled by machine learning algorithms. In this work, we take inspiration from AMLAS, but adapt it to focus on emergence as the driver for complexity, rather than learning.

3 The AERoS Process

This section discusses the six main stages of AERoS targeting autonomous robot swarms. AERoS is iterative by design, and the assurance activities are performed in parallel to EB development (see Fig. 1). For each stage, we describe its inputs and outputs, main assurance activities and their associated artefacts.

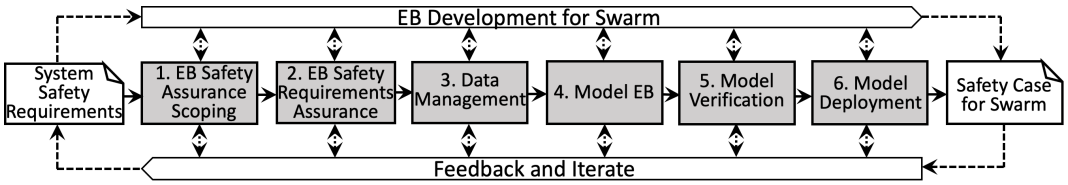

Figure 1: The AERoS process with the six stages adapted from AMLAS.

3.1 Stage 1: EB Safety Assurance Scoping

Stage 1 contains two activities which are performed to define the safety assurance scope for the swarm (see Fig. 2).

Activity 1. Define Assurance Scope for the EB Description and Expected Output The goal of Activity 1, which has four inputs [A–D] (Fig. 2), is to define the safety assurance scope for the EB and expected output. The output of this activity is the Safety Requirements Allocated to the Swarm [E]. The requirements defined in this stage are independent of any EB technology, which reflects the need for the robot swarm to perform safely regardless of emergence.

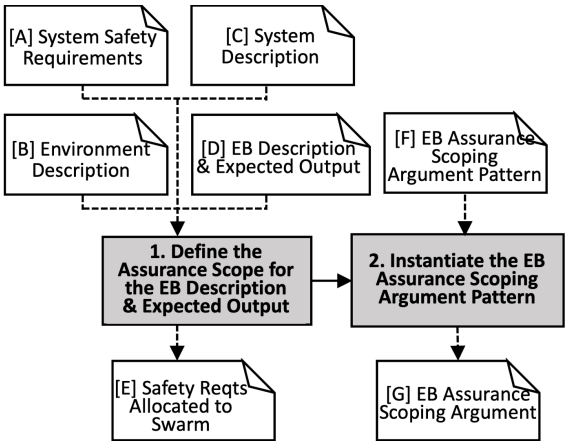

Figure 2: Stage 1: The AERoS emergent behaviour assurance scoping process.

[A] System Safety Requirements: The system safety assessment process generates the safety requirements of the swarm, and covers the identification of hazards (e.g. the blocking of critical paths in the cloakroom) and risk analysis. Figure 3 illustrates how individual robot failures propagate through the neighbourhood to swarm-level hazards: we can then derive safety requirements in the form of concrete failure conditions at the level of the whole swarm which capture, implicitly, all levels of the swarm. Although this has been illustrated as a simplified linear chain of events, in reality this represents a complex sequence which can be difficult to distil into distinct events and causes.

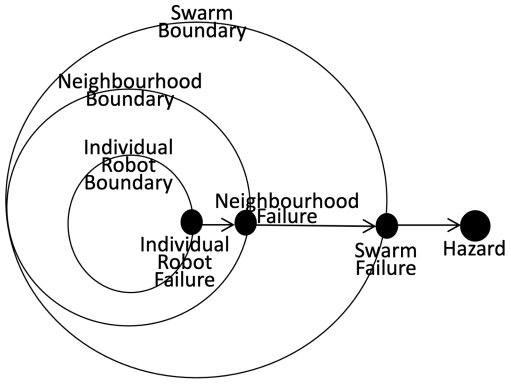

Figure 3: Failure conditions in a swarm adapted from DO-178C and AMLAS.

[B] Environment Description: It is essential to consider the system environment when allocating safety requirements to the swarm. In the cloakroom, a swarm of robotic agents collects and delivers jackets, which are stored in small box-like containers. The agents are required to navigate a public space between collection and delivery points.

[C] System Description: In the cloakroom, we can consider three inputs: sensor availability, neighbourhood data (used because there is no access to global data in real-world deployments), and swarm parameters (see Fig. 4). The *sensors* available to agents can be cameras and laser time-of-flight sensors. The neighbourhood data of the swarm can be specified through the communication systems available to agents, in this case Bluetooth. Through the use of this short-range communication, agents can access neighbourhood data, such as approximate position or current state of local agents. As for the swarm-level parameters, we can consider options specified by a user, that is, the number of agents deployed, and the maximum speed of agents. Once defined, the three inputs are then fed to the individual agents to instruct their behaviour. This behaviour enacted by multiple agents then produces a swarm-level EB as the individuals interact with one another and their environment.

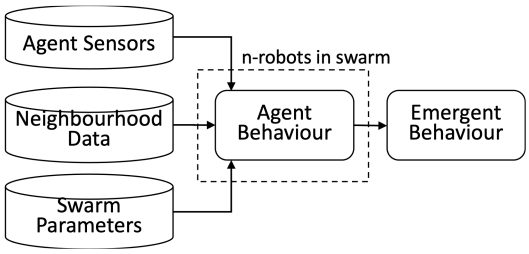

Figure 4: Inputs fed into individual agent behaviour producing overall swarm emergent behaviour.

[D] EB Description and Expected Output: By expected output, we refer to the gains that can arise from the system by deploying multiple agents. In the cloakroom, the output is a collaborative system capable of collecting, sorting, and delivering jackets in a public setting. To achieve this, the EB of the system needs to arise from the available behaviours of the individual agents, their interactions, and the constraints outlined in the system description.

Activity 2. Instantiate EB Assurance Scoping Argument Pattern Each stage of the AERoS process includes an activity to instantiate a safety argument pattern based on the evidence and artefacts generated in that stage. *Argument patterns* [8], which are modelled using the Goal Structuring Notation, can be used to explain the extent to which the evidence supports the relevant EB safety claims. In Activity 2, we use the artefacts generated from Stage 1 (i.e. [A–E]) to instantiate the EB Assurance Scoping Argument Pattern ([F] – see Fig. 5). The instantiated argument [G] along with other instantiated arguments resulting from the other five stages of AERoS constitute the safety case for the swarm. The activities to instantiate argument patterns of the other stages follow a very similar pattern so are not shown due to space limitations.

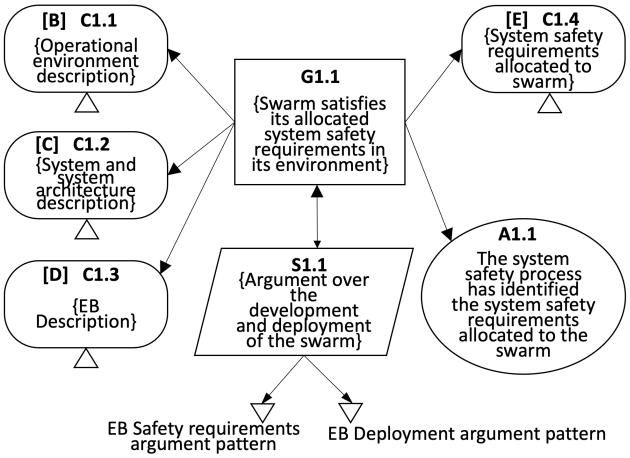

Figure 5: Emergent behaviour safety assurance scoping argument pattern.

3.2 Stage 2: EB Safety Requirements Assurance

Stage 2 contains three activities (Fig. 6), which are performed to provide assurance in EB safety requirements for the swarm.

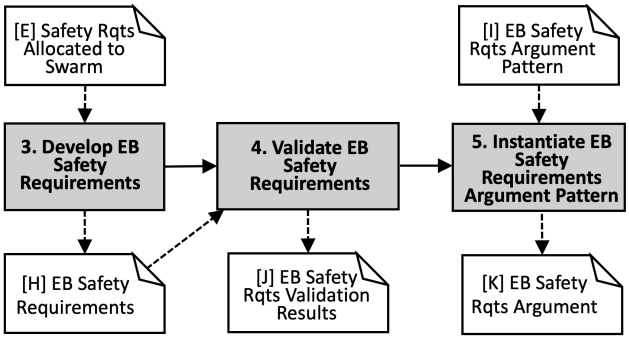

Figure 6: Stage 2: The AERoS emergent behaviour safety requirements assurance.

Activity 3. Develop EB Safety Requirements The input to Activity 3 in Stage 2 is the Safety Requirements Allocated to the Swarm [E]. We define EB safety requirements to specify risk controls for the swarm-level hazards by taking into account the system architecture defined and the operating environment.

In the swarm context, we consider four types of requirements: *performance*, *adaptability*, *human safety*, and *environment*. In particular, the environment requirements capture the need for the system to be robust to variation in the operative space. We consider several safety metrics under each requirement category: (i) performance: low-impact and high-impact collisions (the swarm should operate below a *critical number* of such collisions); (ii) adaptability: percentage of swarm stationary outside of the delivery site, number of stationary agents, time since last agent moved; (iii) human safety: velocity or average velocity of agents, swarm size, rate of humans encountered, proximity to humans; (iv) environment: sum of objects in an area (the density of objects in the environment should not block swarm operation). As the robot swarm is composed of many agents, there is potential for a large number of faults to occur at any given time [13]. This motivates three further sub-categories for each of the performance, adaptability and human-safety requirements: *faultless operations*, *failure modes (graceful degradation)*, and *worst case*. *Graceful degradation* refers to the acceptable level of faults, their impact, and how the system should react when faults occur. *Worst case* accounts for the least acceptable impact the system should experience and the means to avoid it. A key output of Activity 3 is [H], which describes the EB Safety Requirements relating to: performance, adaptability, and environment (see Table 1), as well as human safety (Table 2). These example requirements have been generated following three main considerations: the hazards for each safety requirement type, the metrics [13] available to assess these hazards, and the realistic thresholds [14] given the specification of the system.

Activity 4: Validate EB Safety Requirements The required input to Activity 4 is the EB Safety Requirements [H]. These are validated by both review and simulation. Firstly, the requirements derived for the cloakroom have been reviewed by a safety-critical systems engineering expert to ensure that the specified EB safety requirements for the swarm will deliver its intended safe operation. Secondly, we validated all safety requirements (excepting RQ3.5 from Table 1) for the cloakroom system using the Gazebo 3D simulator. This simulation is an exact replica of the 4m x 4m lab environment used for hardware implementation (see Fig. 7). In **Activity 5**, the artefacts generated in this stage are used to instantiate the EB Safety Requirements Argument Pattern [I].

3.3 Stage 3: Data Management

When designing EB, input data for training an algorithm comes from local sensing of individual agents, both onboard the agent itself and in its local environment. The activities and outputs in this stage take into account the complexities of interactions between multiple agents.

Activity 6. Define Data Requirements In our adaptation of Activity 6, we take the EB Safety Requirements [H] outlined in Stage 2 as an input (see Fig. 8). These safety requirements guide the data requirements in this activity, feeding into the data specification outlined here. We split the data requirement outputs into two multi-agent focused requirements: [L.0] Data Type Requirements and [L.1] Data Availability Constraints.

[L.0] Data Type Requirements: This element focuses on the *relevance*, *completeness*, *accuracy*, and *balance* of the information that will be used to construct the swarm behaviour, and subsequently, to test the EB of the system before

Table 1: Examples of performance, adaptability, and environmental safety requirements for the cloakroom scenario.

RQ	Performance Requirements
1.1	The swarm *shall* experience $<$ **1 high-impact (V $>$ 0.5m/s)** collisions across **a day** of **faultless** operation.
1.2	The swarm *shall* experience $<$ **0.1%** increase in **high-impact** collisions across **a day's** operation with **full communication faults** occurring in **10%** of the swarm.
1.3	The swarm *shall* experience $<$ **0.1%** increase in **high-impact** collisions across **a day's** operation with **half-of-wheels motor faults** occurring in **50%** of the swarm.
1.4	The swarm *shall* experience $<$ **2 high-impact (V $>$ 0.5m/s)** collisions across **a day** of **faulty** operation.
1.5	The swarm agents *shall* **weigh** $<$ **3kg** and shall have **acceleration** $<$ **4m/s** so that the **maximum collision force** in the swarm is within acceptable bounds.
1.6	The swarm agents *shall* only carry objects of **weight** $<$ **2kg**.
	Adaptability Requirements
2.1	The swarm *shall* have $<$ **10%** of its agents **stationary*** outside of the **delivery site** at a given time. *Assumption: Agents are considered stationary once they have not moved for $>$ **10 seconds**.
2.2	All agents of the swarm *shall* move at least every **100 seconds** if outside of the **delivery site**.
2.3	The swarm *shall* experience $<$ **10%** increase in the **number of stationary agents** at any time with **half-of-wheels motor faults** occurring in **50%** of the swarm.
2.4	The swarm agents *shall* experience $<$ **10%** increase in **stationary time** with **half-of-wheels motor faults** occurring in **50%** of the swarm.
2.5	The swarm *shall* experience $<$ **10%** increase in **number of stationary agents** at any given time with **full communication faults** occurring in **10%** of the swarm.
2.6	The swarm agents *shall* experience $<$ **10%** increase in **stationary time** with **full communication faults** occurring in **10%** of the swarm.
2.7	The swarm *shall* have $<$ **20%** of its agents **stationary*** outside of the **delivery site** at a given time. *Assumption: Agents are considered stationary once they have not moved for $>$ **10 seconds**.
	Environmental Requirements
3.1	The swarm *shall* perform as required in environmental density levels **0-4 $\mathbf{p}_o$ of objects** (sum of boxes and agents per m^2) in the environment.
3.2	The swarm *shall* perform as required when **floor incline** is **0-20 degrees**.
3.3	The swarm *shall* perform as required in a **dry environment**.
3.4	The swarm *shall* perform as required in **smooth-floored environments** with step increases no greater than **0.5cm**.
3.5	The swarm *shall* only operate in **environments where humans have devices that identify the human's location** to the swarm agents.

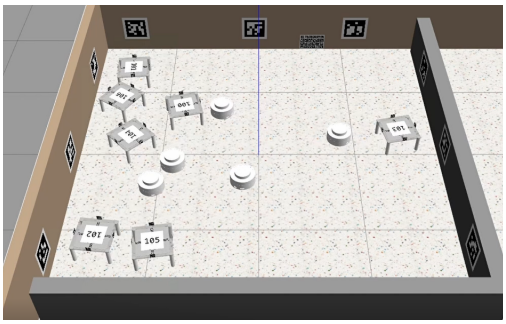

Figure 7: 3D simulation created to validate several emergent behaviour safety requirements.

deployment. The *relevance* of the data used in the development of the EB specifies the extent to which the test environment must match the intended operating domain of the robot swarm. The *completeness* of the data specifies the conditions under which we test the behaviour, that is, the volume of experiments or tests that will be run, the variety of tests executed, and the diversity of environments expected to be used in the testing process. The aim is to cover a representative sample of conditions for testing. *Accuracy* in this context relates to how well the data captures the

Table 2: Examples of human-safety requirements for the cloakroom scenario.

RQ	Human-Safety Requirements
4.1	The swarm agents *shall* travel at speeds of less than **0.5m/s** when within **2m** distance of a **trained human** (a worker who has received relevant training).
4.2	The swarm agents *shall* travel at speeds of less than **0.25m/s** when within **3m** distance of an **attendee**.
4.3	The swarm agents *shall* only come within **2m** distance of a **human** $<$ **10** times collectively across **1000 seconds** of **faultless** operations.
4.4	The swarm *shall* only allow $<$ **5 agents** to request intervention from a **trained human** at a given time.
4.5	A **trained human** *shall* monitor **5-20 agents** at a given time.
4.6	The swarm *shall* only allow **1 agent** to request input from an **attendee** at a given time.
4.7	An **attendee** *shall* receive information from $<$ **5 agents** of the swarm at a given time.
4.8	The swarm *shall* experience $<$ **10%** increase in **human encounters** across **1000 seconds** of operation with **full communication faults** occurring in **10%** of the swarm.
4.9	The swarm *shall* experience $<$ **10%** increase in **human encounters** across **1000 seconds** of operation with **half-of-wheels motor faults** occurring in **50%** of the swarm.
4.10	The swarm agents *shall* only come within **2m** distance of a **human** $<$ **20** times collectively across **1000 seconds** of **faulty** operations.

parameter space defining the performance of the robot swarm. For example, an accurate dataset for what constitutes a delivery in a logistics scenario [15] should track the footprint of a deliverable to ensure it is well-positioned in the delivery zone (RQ7.1). *Balance* refers to the evenly distributed trials executed in the testing process of the EB algorithm. By considering balance, we expect the number of tests conducted for failure modes or environments to be justified, ensuring that there is not an unrealistic bias in testing towards a particular scenario. See Table 3 for examples of data requirements relating to relevance, completeness, accuracy, and balance.

[L.1] Data Availability Constraints: With the introduction of multiple agents comes the issue of data availability. Distributed communication is a key feature found in emergent systems. As such, it is crucial to define how much information each agent is expected to hold, how easily data may transfer between agents, and across what range agents should be able to transfer information between one another. Feasible constraints include [14]: (i) *storage capacity:* the swarm agents *shall* have a maximum of 2 GB of information stored on board at any time; (ii) *available sensors:* the swarm agents *shall* only have access to environmental data deemed feasibly collectable by radially positioned cameras and laser time-of-flight sensors; (iii) *communication range:* the swarm agents *shall* only have access to other agent data when within communications range of 5 metres; and (iv) *operator feedback:* the swarm agents *shall* only share information with non-agents (e.g. operator terminal) when within communications range of 5 metres.

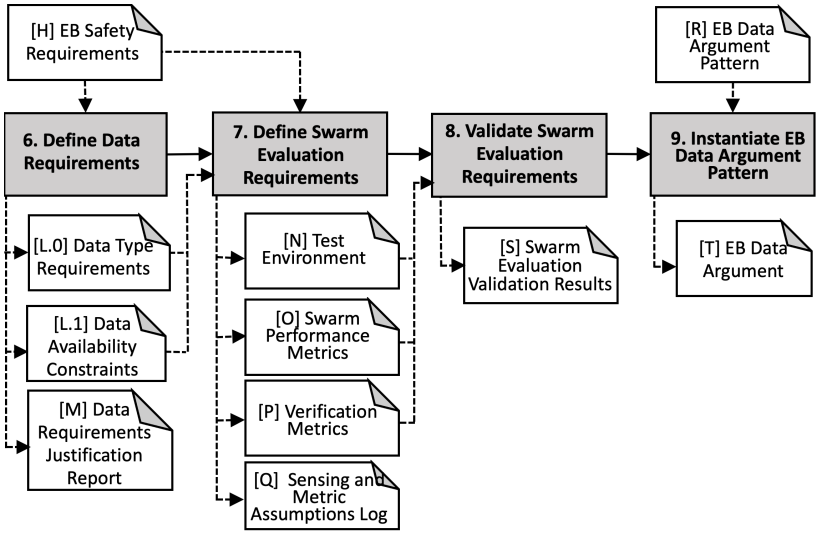

Figure 8: Stage 3: The AERoS data management process.

Table 3: Examples of requirements for output [L.0].

RQ	Relevance Requirements Examples
5.1	All simulations *shall* include environments with ranges of incline between 0-20°.
5.2	All simulations *shall* be conducted in a dry environment.
	Completeness Requirements Examples
6.1	All simulations *shall* be repeated to include occurrences of faults representative of full communication faults.
6.2	All simulations *shall* be repeated a sufficient number of times to ensure results are representative of typical use.
6.3	All simulations *shall* be repeated in multiple environments representative of those expected in real-world use of the system.
	Accuracy Requirements Examples
7.1	All boxes *shall* only be considered 'delivered', if all four of the boxes' feet are positioned within the delivery zone.
7.2	All boxes *shall* only be considered 'delivered', once they are no longer in direct contact with a swarm agent.
	Balance Requirements Examples
8.1	All simulations *shall* be repeated so as to obtain representative evaluations for each possible mode of failure (defined under performance, adaptability, and human-safety requirements in Stage 2).
8.2	All simulations *shall* be repeated equally across all test environments.

[M] Data Requirements Justification Report: This report is an assessment of the data requirements, providing analysis and explanation for how the requirements and constraints ([L.0] and [L.1]) address the EB Safety Requirements [H].

Activity 7. Define Swarm Evaluation Requirements Taking the outputs [L.0] and [L.1] from Activity 6, the evaluation requirements consider how the EB of the swarm will be assessed, specifying the testing environment and the metrics to be used to assess the test results.

[N] Test Environment: This takes into consideration the requirements specified in Activity 6, and defines the environment in which the EB will be tested. In most cases this will be multiple simulation environments featuring diverse sets of the terrain, environmental conditions, and obstacle configurations. There may also be instances in which this test environment is specified as a physical environment operating under laboratory conditions, with a hardware system acting as a test bed to observe designed behaviours.

[O] Swarm Performance Metrics: This output is used to quantify how well the system is performing. While there may be multiple performance metrics, these metrics should be defined with respect to the primary function of the robot swarm. Metrics that might feature in this output could include: the delivery rate in a logistics scenario, the rate of area coverage in an exploration task, or the response time in disaster scenarios.

[P] Verification Metrics: These metrics should be derived from the EB Safety Requirements [H] specified in Stage 2. They are intended to be used as the criteria for success within the verification process. For example, swarm density, which is used in verifying environmental safety specifications such as RQ3.1, maximum collision force experienced by agents, which could be used to verify that the swarm meets performance requirements such as RQ1.1 and RQ1.2, or the current speed of all agents, a metric relating directly to the human-safety requirements RQ4.1 and RQ4.2. Identifying [P] early, ideally during the requirements assurance stage, allows consideration of [P] during the design and development of the swarm to facilitate verification.

[Q] Sensing and Metric Assumptions Log: This log serves as a record of the details and decisions made in Activities 6 and 7. It should contain details of the choices made when producing the Test Environment [N], Swarm Performance Metrics [M], and the Verification Metric [P].

Activity 8. Validate Evaluation Requirements Taking into account outputs [N], [O], and [P] from Activity 7, this activity aims to validate these components with respect to the requirements specified in Activity 6. Should any discrepancies exist between the data requirements and the evaluation requirements, they should be fully justified and recorded in the output Swarm Evaluation Validation Results [S]. The artefacts generated in this stage are used to instantiate the EB Data Argument Pattern [R] in **Activity 9**.

3.4 Stage 4: Model Emergent Behaviour

In the design of an EB algorithm, the challenge is in selecting behaviours at the individual level of the agents which give rise to the desired EB at the swarm level. In our adaptation of AMLAS for the robot swarm, we step away from the machine learning paradigm to allow consideration for all possible optimisation algorithms which may attain the target EB.

Activity 10. Create EB Algorithm This can be nature inspired, hand designed, or evolved from a relatively simple set of instructions for individual behaviour, which takes into account agent-to-agent and environmental interactions [16]. These instructions when given to a large number of agents, create a synergistic behaviour for the swarm that is more powerful than the sum of the individual agent's performance. The EB algorithm is engineered at the level of the individual agent behaviours for the Test Environment output [N] from Stage 3. The resultant EB must meet the Safety Requirements [H] defined in Stage 2 (see Fig. 9). In the cloakroom case study, the target EB for the swarm must ensure that items are stored and retrieved by individuals whilst meeting all requirements specified. For example, performance requirements RQ1.1 and RQ1.2 specify an upper bound on the low/high-impact collisions that a swarm shall experience in a given time frame. These requirements may be fulfilled by constraining the maximum velocity of individual robots or by ensuring that a robot has one or more sensory devices, such as a camera, enabling it to detect obstacles. The key output from this activity is the Candidate EB [U] for testing.

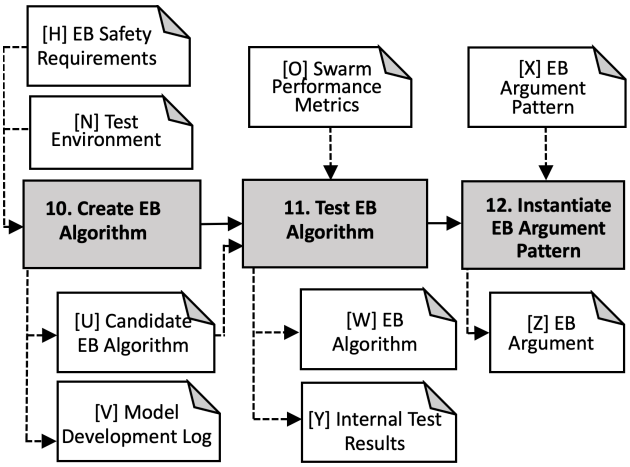

Figure 9: Stage 4: The AERoS model learning process.

[V] Model Development Log: This should log the rationale in the design process of the EB algorithm, in particular how all Safety [H] and Data Type Requirements [L.0] have been met given the Data Availability Constraints [L.1].

Activity 11. Test EB Algorithm In this activity, the candidate EB will be tested against the Swarm Performance Metrics [O] produced in Stage 3. Testing ensures that the EB performs as desired with respect to the defined metrics and in the case where performance passes accepted thresholds, the EB Algorithm [W] will be produced as the output of the activity.

[Y] Internal Test Results: This output provides a degree of transparency in the testing procedure as the results may be further examined to ensure tests have run correctly. In **Activity 12**, the artefacts generated in this stage are used to instantiate the EB Argument Pattern [X].

3.5 Stage 5: Model Verification

Activity 13. Verify EB The inputs to the verification process are the EB Safety Requirements [H], Verification Scenarios (Test Generation) [P], and the EB Algorithm [W] (see Fig. 10). The verification method and assessment process within that method will be largely determined by the specifics of the safety requirements. Some safety specifications lend themselves towards certain assessment methods due to the scenarios they prescribe. For example, to assess that the robot swarm meets the requirements for performance given a motor-fault occurrence (see RQ1.3), it may be easier to realise this in physical or simulation-based testing approaches rather than attempting to construct a formal model of robot behaviour given the complex physical dynamics of a faulty wheel.

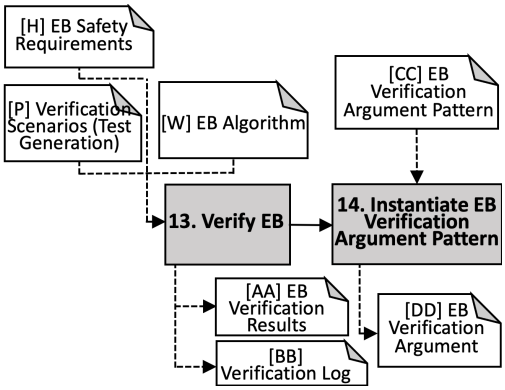

Figure 10: Stage 5: The AERoS verification process.

However, when considering the adaptability requirements, a formal, probabilistic verification technique of the EB Algorithm [W] is more suitable. For example, in RQ2.1, analysis using a probabilistic finite state machine of the swarm behaviour could identify the dwell period within states. Monitors could be used to observe when agents enter a stationary state, for example, $\texttt{agent\_velocity=0} \wedge \texttt{t\_counter} \geq 100$, and identify if time within that state exceeds some fixed value, and ascertain a probabilistic value to this metric.

[P] Verification Scenario (Test Generation): In most cases there will be multiple, valid verification scenarios (test cases) applicable for each of the safety specifications. A 'good test case' must be *effective* at finding defects, *efficient* in minimising the number of tests required, use resources *economically* and be *robust* to system changes [17]. An example of a test case could include a scenario of the swarm in a hazardous environment where too many boxes create an obstacle.

Verification Results [AA] from individual assessments form entries in the Verification Log [BB]. The Verification Log identifies assessments where assurance of the EB Algorithm [W] is acceptable with respect to the Safety Requirements [H] and can be used as a set of evidence for building an assurance case. The artefacts generated in this stage are used to instantiate the EB Verification Argument Pattern [CC] in **Activity 14**.

3.6 Stage 6: Model Deployment

Activity 15. Integrate EB With the EB verified, the next step is to take the EB Algorithm [W], System Safety Requirements [A], Environment Description [B], and System Description [C] and integrate the EB with the system to be deployed (see Fig. 11).

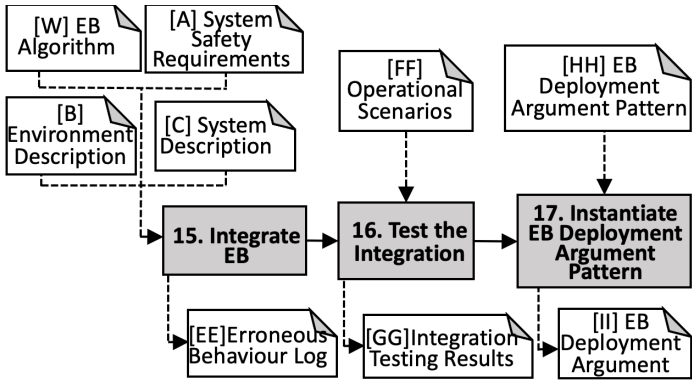

Figure 11: Stage 6: The AERoS model deployment assurance process.

In this activity, we use the inputs to this stage to inform the implementation of the EB and anticipate errors we might expect in the interactions between agents and the overall EB. Despite the rigorous validation and testing conducted in previous stages, there will still be a gap between the test environment and the intended, everyday-use, deployed

scenario. The output, [EE] Erroneous Behaviour Log, captures these anticipated gaps between testing and reality and the differences in behaviour that may surface.

Activity 16. Test the Integration Once the initial integration is complete, the physical implementation should undergo additional testing in which the system will be observed in multiple operational scenarios, as specified in [FF].

[FF] Operational Scenarios: These scenarios should reflect the environment descriptions specified in [B], offering real-world situations to examine the behaviour of the integrated system. The testing of the integrated system in these true-to-operation environments should be conducted in a safe manner, ensuring that the entire multi-agent system can be shut down in an emergency. In the cloakroom, an example of [FF] may take the form of a deployment of agents in a controlled storage area that will not interfere with emergency services.

[GG] Integration Testing Results: Results from the integration testing will be reported here, detailing how the system performs against the EB Safety Requirements [H] specified in Stage 2. The artefacts generated in this stage are used to instantiate the EB Deployment Argument Pattern [HH] in **Activity 17**.

4 Discussion and Future Work

Using AMLAS [8] as a foundation, we have produced the six-stage development process AERoS. This process acts as guidance for those looking to construct swarm robot systems, particularly those that exhibit emergent behaviour through environmental and agent-to-agent interaction. The stages of AERoS break down the design of these systems to ensure that fundamental safety requirements are adhered to, even in instances of system degradation and compounded failures that should be expected, and managed, in robot swarms. We achieve this with an approach that allows for iteration of and feedback to the previous stages as issues of safety are encountered and investigated. We combine this iteration with repeated specification at each stage, observing the issue of safety through the lens of: data, modelling/behaviour design, verification, and deployment.

While the iterative nature of AERoS is a key advantage, some limitations have been identified. First, the scope of this work has been limited to investigating inherent swarm qualities and the emergent properties that arise from these. However, one can expand on this, and consider adaptation of individual robots through techniques such as machine learning (e.g. by applying AMLAS). Second, we can broaden the evaluation by considering additional swarm use cases (e.g. monitoring fires in a natural environment, and also a social swarm), and by providing a worked example of the entire AERoS process.

While the focus of the AERoS process is to ensure the safety assurance of EB in swarms, the trustworthiness of an AS can be dependent on many factors other than safety. These include consideration of ethics, and governance and regulation of AS design and operation. In future work, we intend to build on Porter et al.'s [18] Principle-based Ethical Assurance Argument for AI and Autonomous Systems and develop ethics requirements for swarm robots around the ethical principles of beneficence, non-maleficence, respect for autonomy, and justice. In addition to ethics requirements, we intend to introduce regulatory requirements into the consideration of AS specification. In particular, we observe the work of Macrae's [19] Structural, Organisational, Technological, Epistemic, and Cultural (SOTEC) framework to help us identify sources of socio-technical risk in Autonomous and Intelligent systems. Viewing regulatory requirement analysis from a socio-technical perspective allows us to move away from a purely technical conception of requirements, and helps us design AS that better fit the organisation and operators' work in which safety considerations are meaningful within the wider system and operational context. The relevance of SOTEC for crafting regulatory requirements for the swarms in the cloakroom as a safety assurance mechanism will be described in a future paper.

Acknowledgements

The authors would like to thank Alvin Wilby, John Downer, Jonathan Ives, and the AMLAS team for their fruitful comments. The work presented has been supported by the UKRI Trustworthy Autonomous Systems Node in Functionality under Grant EP/V026518/1. I.H. is supported by the Assuring Autonomy International Programme at the University of York.

References

[1] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics*, volume 3342 of *LNCS*, pages 10–20. Springer, 2005.

[2] Dhaminda B. Abeywickrama, Amel Bennaceur, Greg Chance, Yiannis Demiris, Anastasia Kordoni, Mark Levine, Luke Moffat, Luc Moreau, Mohammad Reza Mousavi, Bashar Nuseibeh, Subramanian Ramamoorthy, Jan Oliver Ringert, James Wilson, Shane Windsor, and Kerstin Eder. On specifying for trustworthiness. 2022.

[3] International Organization for Standardization. ISO/IEC/IEEE 24765:2017 Systems and software engineering — Vocabulary. Online, 2017.

[4] Betty H. C. Cheng, Kerstin I. Eder, Martin Gogolla, Lars Grunske, Marin Litoiu, Hausi A. Müller, Patrizio Pelliccione, Anna Perini, Nauman A. Qureshi, Bernhard Rumpe, Daniel Schneider, Frank Trollmann, and Norha M. Villegas. *Using models at runtime to address assurance for self-adaptive systems*, pages 101–136. Springer International Publishing, Cham, 2014.

[5] Kevin Forsberg and Harold Mooz. The relationship of systems engineering to the project cycle. *Engineering Management Journal*, 4(3):36–43, September 1992.

[6] Yan Jia, John McDermid, Tom Lawton, and Ibrahim Habli. The role of explainability in assuring safety of machine learning in healthcare. *IEEE Transactions on Emerging Topics in Computing*, 10(4):1746–1760, 2022.

[7] Michael Fisher, Viviana Mascardi, Kristin Yvonne Rozier, Bernd-Holger Schlingloff, Michael Winikoff, and Neil Yorke-Smith. Towards a framework for certification of reliable autonomous systems. *Autonomous Agents and Multi-Agent Systems*, 35(8):65, 2021.

[8] Richard Hawkins, Colin Paterson, Chiara Picardi, Yan Jia, Radu Calinescu, and Ibrahim Habli. Guidance on the assurance of machine learning in autonomous systems (AMLAS). Guidance Version 1.1, University of York, March 2021.

[9] Simon Jones, Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. Distributed situational awareness in robot swarms. *Advanced Intelligent Systems*, 2(11):2000110, 2020.

[10] John Rushby. Runtime certification. In Martin Leucker, editor, *Runtime verification*, pages 21–35, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[11] A. Winfield, S. Booth, L. A. Dennis, T. Egawa, H. Hastie, N. Jacobs, R. I. Muttram, J. I. Olszewska, F. Rajabiyazdi, A. Theodorou, M. A. Underwood, R. H. Wortham, and E. Watson. IEEE P7001: A proposed standard on transparency. *Frontiers in robotics and AI*, 8:225, 2021.

[12] Fateh Kaakai, Konstantin Dmitriev, Sridhar Adibhatla, Elgiz Baskaya, Emanuele Bezzecchi, Ramesh Bharadwaj, Barclay Brown, Giacomo Gentile, Corinne Gingins, Stephane Grihon, and Christophe Travers. Toward a machine learning development lifecycle for product certification and approval in aviation. *SAE Int. J. Aerosp.*, 15(2), May 2022.

[13] Suet Lee, Emma Milner, and Sabine Hauert. A data-driven method for metric extraction to detect faults in robot swarms. *IEEE Robotics and Automation Letters*, 7(4):10746–10753, 2022.

[14] Simon Jones, Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. DOTS: An open testbed for industrial swarm robotic solutions, 2022.

[15] Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. Stochastic behaviours for retrieval of storage items using simulated robot swarms. *Artificial Life and Robotics*, 27(2):264–271, 2022.

[16] Simon Jones, Matthew Studley, Sabine Hauert, and Alan Winfield. *Evolving behaviour trees for swarm robotics*, pages 487–501. Springer International Publishing, Cham, 2018.

[17] Mark Fewster and Dorothy Graham. *Software test automation effective use of test execution tools*. 1999.

[18] Zoe Porter, Ibrahim Habli, and John Alexander McDermid. A principle-based ethical assurance argument for AI and autonomous systems, March 2022.

[19] Carl Macrae. Learning from the failure of autonomous and intelligent systems: Accidents, safety, and sociotechnical sources of risk. *Risk analysis*, 2021.