AERoS: Assurance of Emergent Behaviour for use in Autonomous Robotic Swarms

Dhaminda B. Abeywickrama†[1][0000−0002−4423−0284],
James Wilson†[1][0000−0002−0758−6732], Suet Lee[1][0000−0002−9196−5769],
Greg Chance[1][0000−0001−5334−370X], Peter D. Winter[1][0000−0003−0766−6297],
Arianna Manzini[1][0000−0001−7710−8974], Ibrahim Habli[2][0000−0003−2736−8238],
Shane Windsor[1][0000−0002−7597−4497], Sabine Hauert[1][0000−0003−0341−7306], and
Kerstin Eder[1][0000−0001−9746−1409]⋆

[1] University of Bristol, Bristol, UK
{firstname.lastname}@bristol.ac.uk
[2] University of York, York, UK
ibrahim.habli@york.ac.uk

Abstract. The behaviours of a swarm are not explicitly engineered. Instead, they are an emergent consequence of the interaction of individual agents with each other within the swarm and their environment. This emergent functionality poses a challenge to *assurance*, such as safety assurance. The main contribution of this paper is a process for the safety assurance of emergent behaviour for use 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 guidance using an illustrative case study centred on a robot swarm operating a public cloakroom.

Keywords: Assurance · Safety · Emergent behaviour · Guidance · Swarms.

1 Introduction

Swarm robotics provides an approach to the coordination of large numbers of robots, which is inspired from the observation of social insects [13]. The functionality of a swarm is emergent, and evolves based on the capabilities of the robots and the numbers of robots used. The overall behaviours of a swarm are not explicitly engineered in the system, but they are an emergent consequence of the interaction of individual agents with each other and the environment [1], and this poses a challenge to ensure their *assurance*.

According to the ISO standard for systems and software engineering vocabulary [11], *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

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

(V&V), and certification. Assurance criteria for autonomous systems (AS) include both functional and non-functional requirements such as safety [2].

A key limitation of the existing standards and regulations of AS is that they do not accommodate the adaptive nature of AS with evolving functionality [4]. They are either implicitly or explicitly based on the V&V model, which moves from requirements through design onto implementation and testing before deployment [6]. However, this model is unlikely to be suitable for systems with the ability to adapt their functionality in operation; 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) [1]. However, although these industry 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 emergent behaviour (EB).

The main contribution of this paper is a process or guidance for the safety assurance of EB for use in autonomous robotic swarms (AERoS), adapted from the guidance on the Assurance of Machine Learning for use in Autonomous Systems (AMLAS) [5] proposed for machine learning systems. 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 generic and therefore can be applied to any swarm type (e.g. grounded, airborne). In this paper, we explore it using an illustrative case study centered on a grounded robot swarm operating in a public cloakroom. Let us consider a company that organises bespoke events with 50 to 10000 attendees [7]. The company automates their cloakroom by deploying a swarm of robots to assist attendees (customers) to deposit, store, and deliver their belongings (e.g. jackets) [7]. 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 with evolving functionality are considered to follow a much more iterative life-cycle compared to the conventional V&V model. Thus, there is a need for new standards and assurance processes that extend beyond design time and allow continuous certification at runtime [12]. In this context, there have been several standards and guidance introduced by several 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 [14], P7007, P7010). There are several standards and guidance related to machine learning in aeronautics, automotive, railway and industrial domains [8], for example the AMLAS process [5], 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. None of the aforementioned standards and guidance targets swarm systems.

In this work we used AMLAS [5] 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, and the assurance activities are performed in parallel to the development of machine learning component. The process is iterative by design and feedback is used to update previous stages. However, AMLAS specifically targets the safety assurance of a machine learning model for a single system, and not for a collective as considered in our study.

3 The AERoS Process

In this section, we discuss the six main stages of the AERoS process targeting swarm systems. For each stage, we describe its inputs and outputs, main assurance activities and their associated artefacts.

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. 1). In a swarm, individual robots execute simple behaviours, and when performed by a large number of agents, these simple behaviours build to form EB.

Activity 1. Define Assurance Scope for the EB Description and Expected Output The goal of Activity 1, which has four inputs [A–D], is to define the safety assurance scope for the EB description 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 technique or metric, which reflects the need for the swarm system to perform safely regardless of the deployed technology.

[A] System Safety Requirements: The system safety assessment process generates the safety requirements of the swarm, and it covers the identification of hazards (e.g. the blocking of critical paths in the cloakroom) and risk analysis. Figure 2 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

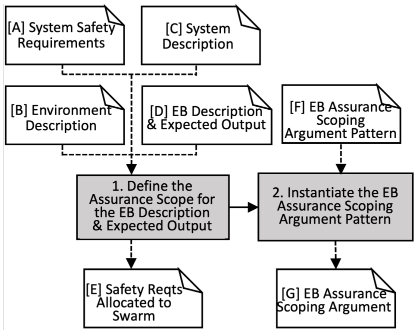
[A] System Safety Requirements
[C] System Description
[B] Environment Description
[D] EB Description & Expected Output
[F] EB Assurance Scoping Argument Pattern
1. Define the Assurance Scope for the EB Description & Expected Output
2. Instantiate the EB Assurance Scoping Argument Pattern
[E] Safety Reqts Allocated to Swarm
[G] EB Assurance Scoping Argument

Fig. 1. Stage 1: The AERoS EB assurance scoping process.

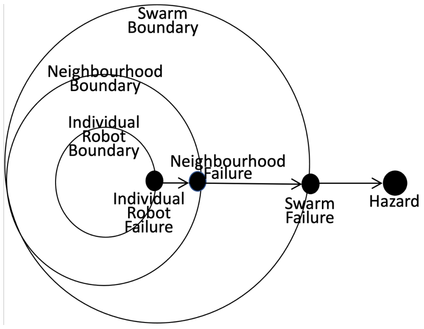

Fig. 2. Failure conditions in a swarm adapted from DO-178C and AMLAS.

simplified linear chain of events, in reality this represents a complex sequence which can be difficult to distil into distinct events and causes.

[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. They use local communication, perception, and data to form an emergent system of navigation which allows them to easily traverse the public space.

[C] System Description: In the cloakroom, we can consider three inputs: sensor availability, neighbourhood data, and swarm parameters (see Fig. 3). The *sensors* available to agents can be: cameras, Bluetooth communication devices, and light detection and ranging systems. 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 of local agents, current behaviour statuses, an approximate history of box movement, and the amount of time deployed. As for the swarm level parameters, we can consider options specified by a user, that is, the number of agents deployed, maximum speed of agents, and the number of agents permitted to be active at a time. 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.

[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 redelivering jackets in a public setting. For this the EB of the system needs to be manually engineered with consideration given to the available sub-behaviours within an agent and the constraints outlined in the system description.

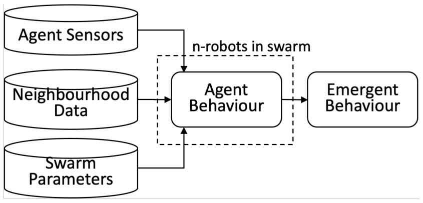

Fig. 3. System description.

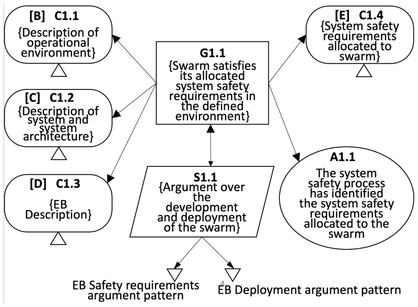
[B] C1.1
{Description of operational environment}
[C] C1.2
{Description of system and system architecture}
[D] C1.3
{EB Description}
G1.1
{Swarm satisfies its allocated system safety requirements in the defined environment}
S1.1
{Argument over the development and deployment of the swarm}
[E] C1.4
{System safety requirements allocated to swarm}
A1.1
The system safety process has identified the system safety requirements allocated to the swarm
EB Safety requirements argument pattern
EB Deployment argument pattern

Fig. 4. EB Safety assurance scoping argument pattern.

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, following AMLAS [5]. *Argument patterns*, 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. 4). 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.

3.2 Stage 2: EB Safety Requirements Assurance

Stage 2 contains three activities (Fig. 5), which are performed to provide assurance in EB safety requirements for the swarm. The scope of this stage is limited to the EB model of the swarm.

Activity 3. Develop EB Safety Requirements The required input to Activity 3 in Stage 2 is the Safety Requirements Allocated to the Swarm [E]. We define EB safety requirements to control the risk of swarm level hazards by taking into account the system architecture defined and 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 performance safety metrics under each requirement category: (i) performance: low-impact and high-impact 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/m^2. As the swarm system is composed of many agents, there is potential for a large number of faults to occur at any given time. This motivates three further sub-categories for each of the

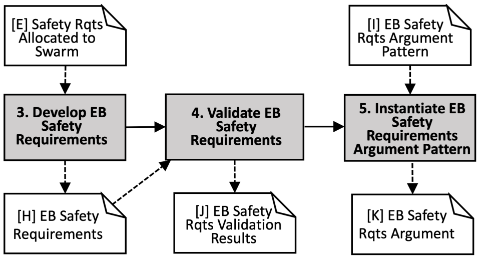
[E] Safety Rqts Allocated to Swarm
3. Develop EB Safety Requirements
4. Validate EB Safety Requirements
5. Instantiate EB Safety Requirements Argument Pattern
[I] EB Safety Rqts Argument Pattern
[H] EB Safety Requirements
[J] EB Safety Rqts Validation Results
[K] EB Safety Rqts Argument

Fig. 5. Stage 2: The AERoS EB safety requirements assurance.

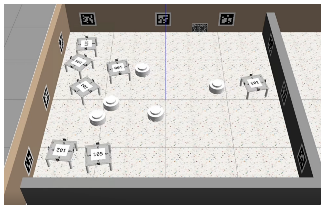

Fig. 6. 3D simulation created to validate several EB safety requirements.

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 those faults are introduced. *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 (see Table 2).

Activity 4: Validate EB Safety Requirements The required input to Activity 4 is the EB Safety Requirements [H]. The EB safety requirements 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, in simulation we validated all safety requirements specified (excepting RQ3.5) for the cloakroom system using the Gazebo 3D simulator. The simulation used is an exact replication of the 4m x 4m lab environment used for hardware implementation (see Fig. 6). 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, data plays a vital role, though one that differs from the machine learning applications AMLAS was initially designed for. In order to address this difference, the following activities and outputs have been adjusted to take into account the swarm behavioural design process and the added complexities that come with multiple agents interacting with one another. Additionally, we specify requirements based on the environments and training parameters which provide the data for training rather than on the datasets themselves.

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. 7). These safety requirements guide the data requirements in this activity,

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 **10% injection** of **full communication fault** to the swarm.
1.3	The swarm *shall* experience < **0.1%** increase in **high-impact** collisions across **a day's** operation with **50% injection** of **half-of-wheels motor faults** to the swarm.
1.4	The swarm *shall* experience < **2 high-impact (V > 0.5m/s)** collisions across **a day** of **faulty** operation.
	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 **number of stationary agents** at any given time with **50% injection** of **half-of-wheels motor faults** to the swarm.
2.4	The swarm agents *shall* experience < **10%** increase in **stationary time** with **50% injection** of **half-of-wheels motor faults** to the swarm.
2.5	The swarm *shall* experience < **10%** increase in **number of stationary agents** at any given time with **10% injection** of **full communication fault** to the swarm.
2.6	The swarm agents *shall* experience < **10%** increase in **stationary time** with **10% injection** of **full communication fault** to 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 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 whereabouts** to the swarm agents.

Table 1. Examples of safety requirements for the cloakroom scenario.

feeding into the data specification that we outline 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 will be subsequently used to test the EB of the system prior to its deployment. The *relevance* of the data used in the development of the

RQ	**Human Safety Requirements**
4.1	The agents in the swarm *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 agents in the swarm *shall* travel at speeds of less than **0.25m/s** when within **3m** distance of a **member of the public**.
4.3	The agents in the swarm *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 a **member of the public** at a given time.
4.7	A **member of the public** *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 **10% injection** of **full communication fault** to the swarm.
4.9	The swarm *shall* experience < **10%** increase in **human encounters** across **1000 seconds** of operation with **50% injection** of **half-of-wheels motor faults** to the swarm.
4.10	The agents in the swarm *shall* only come within **2m** distance of a **human** < **20** times collectively across **1000 seconds** of **faulty** operations.

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

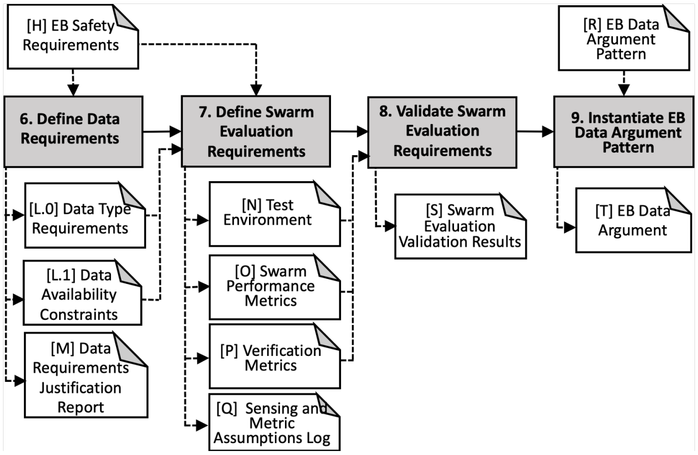
[H] EB Safety Requirements
[R] EB Data Argument Pattern
6. Define Data Requirements
7. Define Swarm Evaluation Requirements
8. Validate Swarm Evaluation Requirements
9. Instantiate EB Data Argument Pattern
[L.0] Data Type Requirements
[L.1] Data Availability Constraints
[M] Data Requirements Justification Report
[N] Test Environment
[O] Swarm Performance Metrics
[P] Verification Metrics
[Q] Sensing and Metric Assumptions Log
[S] Swarm Evaluation Validation Results
[T] EB Data Argument

Fig. 7. Stage 3: The AERoS data management process.

EB specifies the extent to which the test environment must match the intended operating domain into which the model is to be deployed. The *completeness* of the data specifies the conditions under which we test the behaviour algorithm, 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 the parameters defining the performance of the swarm system's primary function. For example, what constitutes a delivery in a logistics scenario, or under what conditions would an area be considered explored in a surveying mission. *Balance* refers to the balance of the 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.

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 fault injections 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.

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

[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: (i) *storage capacity:* the swarm agents shall have a maximum of 2 GB of information stored on board at any point in time; (ii) *available sensors:* the swarm agents shall only have access to environmental data deemed feasibly collectable by radially positioned infrared 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.

[M] Data Requirements Justification Report: This report acts as an assessment of the data requirements, providing analysis and explanation for how the

requirements and constraints (outlined in [L.0] and [L.1]) address the EB Safety Requirements specified in [H].

Activity 7. Define Swarm Evaluation Requirements Taking the outputs [L.0] and [L.1] from Activity 6, the evaluation requirements take into account how the EB of the swarm will be assessed, specifying the testing environment and the metrics comprising 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 swarm system. 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. These metrics are intended to be used as the criteria for success within the verification process. Examples of these metrics and their related safety requirements include: 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 safety 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.

[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 justified appropriately 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 agent which give rise to the desired EB at the swarm level.

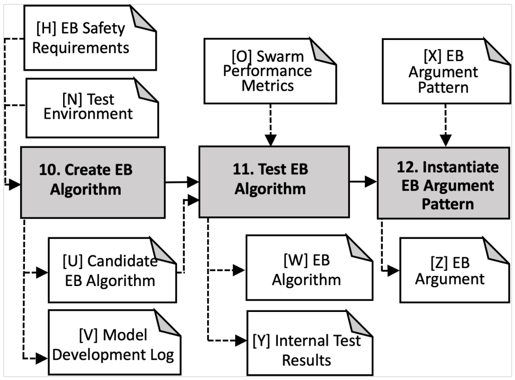
[H] EB Safety Requirements
[N] Test Environment
[O] Swarm Performance Metrics
[X] EB Argument Pattern
10. Create EB Algorithm
11. Test EB Algorithm
12. Instantiate EB Argument Pattern
[U] Candidate EB Algorithm
[V] Model Development Log
[W] EB Algorithm
[Y] Internal Test Results
[Z] EB Argument

Fig. 8. Stage 4: The AERoS model learning process.

In the original AMLAS process, Stage 4 focuses on the creation, testing, and instantiation of a machine learning model for a single system with no consideration given to EB for a collective. 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 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. 8). 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 or infrared sensors, enabling it to detect obstacles. The key output from this activity is the Candidate EB [U] for testing.

[V] Model Development Log: This should log the rationale in the design process of the EB algorithm, in particular how all specified 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. 9). 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 swarm system meets the requirements for performance given a motor fault injection RQ1.3, it may be easier to realise this in physical or simulation-based testing approaches rather than constructing a reliable formal model of robot behaviour given the complex physical dynamics of a faulty wheel.

However, when considering the adaptability requirements, a formal, probabilistic-based 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, `agent_velocity=0` $\wedge$ `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 bugs or defects, *efficient* in minimising the number of tests required, use resources *economically* and be *robust* to system changes [3].

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. 10).

In this activity, we use the inputs to this stage to educate 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.

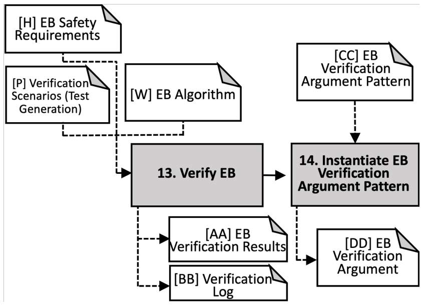
[H] EB Safety Requirements
[P] Verification Scenarios (Test Generation)
[W] EB Algorithm
[CC] EB Verification Argument Pattern
13. Verify EB
14. Instantiate EB Verification Argument Pattern
[AA] EB Verification Results
[BB] Verification Log
[DD] EB Verification Argument

Fig. 9. Stage 5: The AERoS verification process.

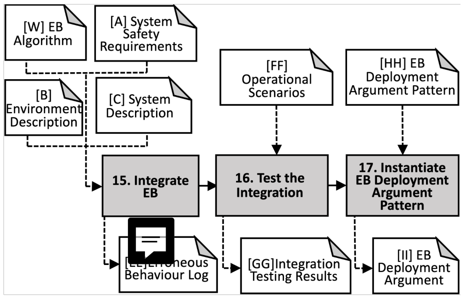
[W] EB Algorithm
[A] System Safety Requirements
[B] Environment Description
[C] System Description
[FF] Operational Scenarios
[HH] EB Deployment Argument Pattern
15. Integrate EB
16. Test the Integration
17. Instantiate EB Deployment Argument Pattern
[EE]Erroneous Behaviour Log
[GG]Integration Testing Results
[II] EB Deployment Argument

Fig. 10. Stage 6: The AERoS model deployment assurance process.

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 operational 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 and ensure 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 small 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 [5] 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 EB 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 swarm robot solutions. 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, a few limitations of this work can be identified. First, the scope of this work has been limited to investigating inherent swarm qualities and the adaptation that arises 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 a social swarm), and 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 [10] 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 [9] 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.

Acknowledgments

The authors would like to thank John Downer, Jonathan Ives, and Alvin Wilby for their fruitful comments. The work presented in this paper has been supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under the grant [EP/V026518/1].

References

1. Abeywickrama, D.B., Bennaceur, A., Chance, G., Demiris, Y., Kordoni, A., Levine, M., Moffat, L., Moreau, L., Mousavi, M.R., Nuseibeh, B., Ramamoorthy, S., Ringert, J.O., Wilson, J., Windsor, S., Eder, K.: On specifying for trustworthiness (2022), http://arxiv.org/abs/2206.11421
2. Cheng, B.H.C., Eder, K.I., Gogolla, M., Grunske, L., Litoiu, M., Müller, H.A., Pelliccione, P., Perini, A., Qureshi, N.A., Rumpe, B., Schneider, D., Trollmann, F., Villegas, N.M.: Using Models at Runtime to Address Assurance for Self-Adaptive Systems, pp. 101–136. Springer International Publishing, Cham (2014)
3. Fewster, M., Graham, D.: Software Test Automation Effective use of test execution tools (1999)
4. Fisher, M., Mascardi, V., Rozier, K.Y., Schlingloff, B.H., Winikoff, M., Yorke-Smith, N.: Towards a framework for certification of reliable autonomous systems. Autonomous Agents and Multi-Agent Systems **35**(8), 65 (2021)

5. Hawkins, R., Paterson, C., Picardi, C., Jia, Y., Calinescu, R., Habli, I.: Guidance on the assurance of machine learning in autonomous systems (AMLAS). Guidance Version 1.1, University of York (Mar 2021)
6. Jia, Y., McDermid, J., Lawton, T., Habli, I.: The role of explainability in assuring safety of machine learning in healthcare (2021). https://doi.org/10.48550/ARXIV.2109.00520
7. Jones, S., Milner, E., Sooriyabandara, M., Hauert, S.: Distributed situational awareness in robot swarms. Advanced Intelligent Systems **2**(11), 2000110
8. Kaakai, F., Dmitriev, K., Adibhatla, S., Baskaya, E., Bezzecchi, E., Bharadwaj, R., Brown, B., Gentile, G., Gingins, C., Grihon, S., Travers, C.: Toward a machine learning development lifecycle for product certification and approval in aviation. SAE Int. J. Aerosp. **15**(2) (May 2022). https://doi.org/10.4271/01-15-02-0009
9. Macrae, C.: Learning from the failure of autonomous and intelligent systems: Accidents, safety, and sociotechnical sources of risk. Risk analysis (2021)
10. Porter, Z., Habli, I., McDermid, J.A.: A Principle-based Ethical Assurance Argument for AI and Autonomous Systems (Mar 2022). https://doi.org/10.48550/ARXIV.2203.15370
11. International Organization for Standardization: ISO/IEC/IEEE 24765:2017 Systems and software engineering — Vocabulary. Online (2017), https://www.iso.org/standard/71952.html
12. Rushby, J.: Runtime certification. In: Leucker, M. (ed.) Runtime verification. pp. 21–35. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
13. Şahin, E.: Swarm robotics: From sources of inspiration to domains of application. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics. pp. 10–20. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
14. Winfield, A., Booth, S., Dennis, L.A., Egawa, T., Hastie, H., Jacobs, N., Muttram, R.I., Olszewska, J.I., Rajabiyazdi, F., Theodorou, A., Underwood, M.A., Wortham, R.H., Watson, E.: IEEE P7001: A proposed standard on transparency. Frontiers in robotics and AI **8**, 225 (2021). https://doi.org/10.3389/frobt.2021.665729