Towards the Application of Swarm Intelligence in Safety Critical Systems

Alan F.T. Winfield[*], Christopher J. Harper[†], Julien Nembrini[‡]

[*]Bristol Robotics Laboratories[1], University of the West of England, Bristol BS16 1QY, UK
e-mail: alan.winfield@uwe.ac.uk
[†]Avian Technologies Ltd., 28 Dumaine Avenue, Stoke Gifford, Bristol BS34 8XH, UK
e-mail: cjharper@avian-technologies.co.uk ; fax: (0117) 328 3960
[‡]Swarm-Intelligent Systems Group, EPFL, 1015 Lausanne, Switzerland
e-mail: julien.nembrini@epfl.ch

Keywords: Swarm intelligence; swarm robotics; dependability; distributed safety-critical systems.

Abstract

Swarm Intelligence provides us with a powerful new paradigm for building fully distributed de-centralised systems in which overall system functionality emerges from the interaction of individual agents with each other and with their environment. Such systems are intrinsically highly parallel and can exhibit high levels of robustness and scalability; qualities desirable in high-integrity distributed systems. Making use of a laboratory based swarm robotic system as a case study, this review paper explores dependability, robustness and reliability modelling in swarm based systems, and argues that there is considerable merit in further investigating their application to distributed safety-critical systems.

1 Introduction

The term Swarm Intelligence, first proposed by Gerardo Beni [2], describes the kind of smart or purposeful collective or co-operative behaviours observed in nature, most dramatically in social insects. The past ten years has seen growing research interest in artificial systems based upon the principles of swarm intelligence. In such systems, individual agents make decisions autonomously, based only upon local sensing and communications (see Bonabeau *et al* [3] [4]).

Artificial systems based upon a swarm of physical mobile robots (hence the term *Swarm Robotics*) have been shown to exhibit very high levels of robustness and scalability. Potential applications for swarm robotics might include a swarm of marine robots that find and then contain oil pollution; a swarm of search-and-rescue robots that enter the ruins of a collapsed building to look for survivors and simultaneously map its interstices; or *in-vivo* nano-bots that seek and isolate harmful cells in the blood streams - a kind of artificial phagocyte. (The latter application is not so far-fetched when one considers the rate of progress in the engineering of genetic circuits [25]).

Swarm robotic systems typically exhibit self-organisation, and the overall system behaviours are an *emergent* consequence of the interaction of individual agents with each other and with their environment. For a formal approach to emergence see Kubik [14]. A distinguishing characteristic of distributed systems based upon swarm intelligence is that they have no hierarchical command and control structure, and hence no common-mode failure point or vulnerability. Individual robots are often very simple, even minimalist, and the overall swarm is intrinsically fault-tolerant since, by definition, it consists of a number of identical units operating and cooperating in parallel. In swarms fault tolerance, in a sense, comes for free; that is without special efforts to achieve fault tolerance by the designer. Contrast this with conventional complex distributed systems that require considerable design effort to achieve fault tolerance.

All of this leads to the question: "Might future distributed systems based upon the swarm intelligence paradigm be suitable for application in safety-critical applications?" The aim of this paper is to address that question and its implications for both swarm robotics and safety-critical systems.

This paper proceeds as follows. First, in order to provide a concrete example for discussion we introduce a case study of a swarm robotic system that has been developed and studied within the Bristol Robotics Laboratories. In section 3 the paper asks the question "can swarm systems be dependable?", and considers how safety properties might be established for such a swarm. Section 4 addresses the question of robustness in swarm robotic systems, and section 5 considers reliability modelling for swarm systems.

2 Case Study: Swarm Containment

As a case study let us consider a swarm robotics approach to physical containment or encapsulation, as shown in figure 1.

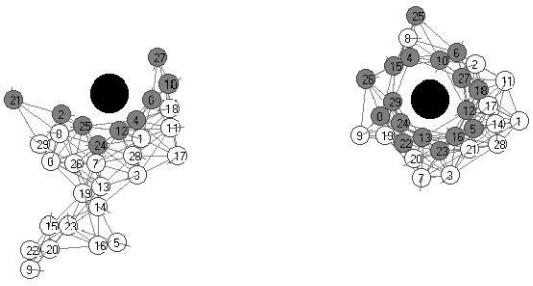

Figure 1: Swarm containment, in progress (l) complete (r)

[1] http://www.brl.ac.uk/

Figure 1 shows a simulated swarm of 30 mobile robots, in a 2D environment, containing or encapsulating an object (the central black circle); on the left containment is in progress; on the right it is complete. The swarm of figure 1 could represent real-world robots containing some chemical or toxic hazard; equally, they could be thought of as future nano-bots in a blood stream. Although in figure 1 some of the robots appear to be touching they are not physically attached to one another. The grey lines joining the robots simply indicate short-range wireless connections. The swarm of figure 1 self-maintains, in fact, a fully connected *ad hoc* wireless network.

The emergent encapsulation behaviour of figure 1 is one of a number of emergent properties of a class of algorithms that we have developed, which make use of local wireless connectivity information alone to achieve swarm aggregation; see [20, 21]. Wireless connectivity is linked to robot motion so that robots within the swarm are wirelessly 'glued' together. This approach has several advantages: firstly the robots need neither absolute or relative positional information; secondly the swarm is able to maintain aggregation (i.e. stay together) even in unbounded space, and thirdly, the connectivity needed for and generated by the algorithm means that the swarm naturally forms an *ad hoc* communications network. Such a network would be an advantage in many swarm robotics applications. The algorithm requires that connectivity information is transmitted only a single hop. Each robot broadcasts its ID and the IDs of its immediate neighbours only, and since the maximum number of neighbours a real robot can have is physically constrained and the same for a swarm of 100 or 10,000 robots, the algorithm scales linearly for increasing swarm size. We have (we contend) a highly robust and scalable swarm of homogeneous and relatively incapable robots with only local sensing and communication capabilities, in which the required swarm behaviours are truly emergent.

The lowest level swarm behaviour is 'coherence' which, in summary, works as follows. Each robot has range-limited wireless communication and, while moving, periodically broadcasts an 'I am here' message (which also contains the IDs of its neighbours). The message will of course be received only by those robots that are within wireless range. If a robot loses a connection to robot *r* and the number of its remaining neighbours still connected to *r* is less than or equal to the threshold *beta* then it assumes it is moving out of the swarm and will execute a 180 degree turn. When the number of connections rises (i.e. when the swarm is regained) the robot chooses a new direction at random. We say that the swarm is *coherent* if any break in its overall connectivity lasts less than a given time constant *C*. Coherence gives rise to the two emergent behaviours of swarm aggregation and a (coherent) connected *ad hoc* wireless network. Each robot also has short-range avoidance sensors and a long-range sensor that detects the object to be contained. When a robot senses the object it sets its *beta* threshold to infinity (the normal value of *beta* is low, typically 2 or 3). This creates a differential motion in the swarm and gives rise to swarm attraction toward the object (*taxis*). Swarm obstacle avoidance and beacon encapsulation behaviours follow naturally. Table 1 summarises the complete set of emergent swarm behaviours.

	Case Study Swarm Behaviours
1	Swarm aggregation
2	Coherent *ad hoc* network
3	Object taxis (attraction)
4	Obstacle avoidance
5	Object encapsulation

Table 1: Summary of emergent swarm behaviours

Our algorithms for coherent swarming of wireless networked mobile robots have been tested extensively in simulation and, rather less extensively, using a fleet of physical laboratory robots. A group of these robots (*Linuxbots*) are shown in figure 2. The real robot implementation does not, however, constitute a real-world application. It is instead an 'embodied' simulation, shown in figure 3, whose main purpose is to verify that algorithms tested in computer simulation will transfer to the real world of non-ideal and noisy sensors and actuators.

Figure 2: Laboratory Linuxbots

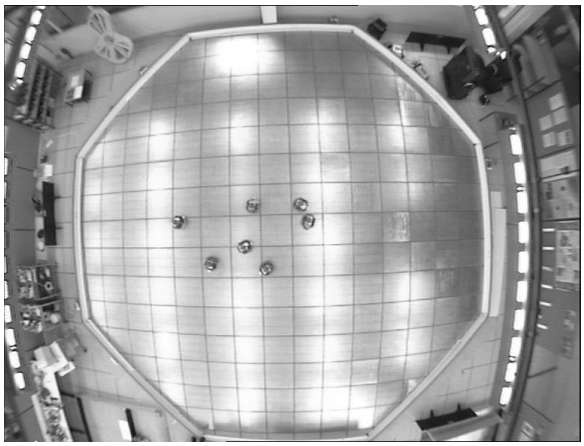

Figure 3: Real-robot swarm tests

3 Can Swarm Systems be Dependable?

From an engineering standpoint the design of complex distributed systems based upon the swarm intelligence paradigm is attractive but problematical. Swarm systems could exhibit much higher levels of robustness, in the sense of tolerance to failure of individual agents, than in conventionally designed distributed systems. However, that robustness comes at a price. Complex systems with swarm intelligence might be difficult to control or mediate if they started to exhibit unexpected behaviours. Such systems would therefore need to be designed and validated for a high level of assurance that they exhibit intended behaviours and *equally importantly* do not exhibit unintended behaviours. It seems reasonable to assert that future engineered systems based on the swarm intelligence paradigm would need to be subject to processes of design, analysis and test no less demanding that those we expect for current complex distributed systems.

Some would argue that the swarm intelligence paradigm is intrinsically unsuitable for application in engineered systems that require a high level of integrity. The idea that overall desired swarm behaviours are not explicitly coded anywhere in the system, but are instead an *emergent consequence* of the interaction of individual agents with each other and their environment, might appear to be especially problematical from a dependability perspective. In a previous paper [22] we argued that systems which employ emergence should, in principle, be no more difficult to validate than conventional complex systems, and that a number of characteristics of swarm intelligence may be highly desirable from a dependability perspective. In that paper we introduced the notion of a 'dependable swarm'; that is a robotic swarm engineered to high standards of design, analysis and test, and therefore able to exhibit high levels of safety and reliability; hence *swarm engineering*. That paper concluded that while some of the tools needed to assure a swarm for dependability exist, most do not, and set out a roadmap of the work that needs to be done before safety-critical swarms become an engineering reality. The present paper follows that roadmap. Probably the most challenging task in dependability assurance [1] is proving the *safety* of a system. Formally, 'safety' is defined as the property of *not exhibiting undesirable behaviours* or, to put it more simply, not doing the wrong thing. The procedure for establishing this property first requires that we identify and articulate all possible undesirable behaviours, by means of a hazard analysis. Of course not all undesirable behaviours are necessarily unsafe (in the ordinary sense of the word 'safe'), as the analysis in section 3.1 will show.

Given a reasonably well understood operational environment there are two reasons for undesirable behaviours: random hardware faults, or systematic (design) errors. Random faults are typically analysed using techniques such as Failure Mode and Effect Analysis (FMEA) [5]. The likelihood that random errors cause undesirable behaviours can be reduced, in the first instance, by employing high reliability components. But systems that require high dependability will typically also need to be fault tolerant, through redundancy for example. This is an important point since swarm engineered systems should, in this respect, offer very significant advantages over conventional complex systems. Two characteristics of swarms work in our favour here. Firstly, simple agents with relatively few rules lend themselves to FMEA, and their simplicity facilitates design for reliability. Secondly, swarms consist of multiple robots and hence *by definition* exhibit high levels of redundancy and tolerance to failure of individual agents. Indeed, robot swarms may go far beyond conventional notions of fault tolerance by exhibiting tolerance to individuals who actively thwart the overall desired swarm behaviour (this is known as Byzantine fault tolerance [7]).

3.1 Case Study FMEA

Let us now consider a Failure Mode and Effect Analysis (FMEA) for the swarm containment case study outlined in section 2. The methodology is straightforward, see Dailey [5]. We attempt to identify all of the possible hazards, which could be faults in robots or robot sub-systems (internal hazards), or environmental disturbances (external hazards). Then, in each case, we analyse the effect of the hazard on each of the overall swarm behaviours. In this way we build up a picture of the tolerance of the swarm to both types of hazard and begin to understand which hazards are the most serious in terms of compromising the overall desired swarm behaviours. FMEA is, at this stage, essentially qualitative. In this paper we consider only internal hazards. External hazards (i.e. communications noise) are investigated in [21].

First we identify the internal hazards. In keeping with the swarm intelligence paradigm our robot swarm contains no system-wide components or structures, thus the only internal hazards that can occur are faults in individual robots. Since, in our case, the robots of the swarm are all identical, then (internal) hazards analysis requires us to consider only the faults that could occur in one or more individual robots, and then consider their effect on the overall swarm behaviours. Table 2 identifies the fault conditions for an individual robot.

Hazard	*Description*
H_1	Motor failure
H_2	Communications failure
H_3	Avoidance sensor(s) failure
H_4	Object sensor failure
H_5	Control systems failure
H_6	All systems failure

Table 2: Internal hazards for a single robot

Table 2 makes the assumption that failures of robot subsystems can occur independently. This is a reasonable assumption, given that our mobile robots are in reality an assembly of complex but relatively self-contained subsystems. Hazard H_1 motor failure, covers the possibility of mechanical or motion-controller failure in one or both of the motors in our differential drive mobile robot, such that the robot is either unable to move at all or can only turn on the spot (which from an overall swarm point of view amounts to the same thing). Hazard H_2 represents a failure of the robot's wireless network communication system such that the robot is

unable to receive or transmit messages. Hazards H_3 and H_4 represent failure of the robot's avoidance and object sensors respectively; the former will render the robot incapable of detecting and hence avoiding robots or environmental obstacles, the latter means that the robot cannot sense the target, i.e. the object to be contained. Hazard H_5 represents a failure of the robot's control system (typically implemented in software). Finally hazard H_6 represents a total failure of the robot; failure of the robot's power supply would, for instance, bring about this terminal condition.

A detailed analysis of the effects of each the hazards enumerated here on the overall swarm behaviours is beyond the scope of this paper, but can be found in [23]. In summary, we find that there are 3 failure effects, listed in table 3.

	Failure effect
E_1	Motor failure anchoring the swarm
E_2	Lost robot(s) loose in the environment
E_3	Robot collisions with obstacles or target

Table 3: Failure Effects

Failure effects E_2 and E_3 in table 3 are self explanatory. E_1 is particularly interesting, and by far the most serious of the three. If one or more robots experience a partial failure such that their motor(s) fail but other sub-systems, in particular network communication remain fully operational (hazard H_1), then the failed robots have the effect of 'anchoring' the swarm and hindering it's motion toward the target. By contrast, complete failure of one or more robot(s) (hazard H_6) will clearly render the robot(s) stationary and inactive. They will be wirelessly disconnected from the swarm and will be treated, by the swarm, as static obstacles to be avoided. Ironically, given that this is the most serious failure at the level of an individual robot, it is the most benign as far as the overall swarm is concerned. Apart from the loss of the failed robots from the swarm, none of the overall swarm behaviours are compromised by this hazard. It is, in fact, the least serious hazard.

Swarm behaviour	H_1	H_2	H_3	H_4	H_5	H_6
Aggregation	-	E_2	-	-	E_2	-
Ad hoc network	-	E_2	-	-	E_2	-
Object taxis	E_1	E_2	-	-	E_1	-
Obstacle avoidance	E_1	E_2	E_3	-	E_1	-
Object containment	E_1	E_2	E_3	-	E_1	-

Table 4: Failure modes and effects

Table 4 shows the swarm fault effects, as defined above, generated by one or a small number of robots with hazards H_1 - H_6, for each of the five emergent swarm behaviours defined in section 2. Table 4 clearly shows that the serious swarm failure effect E_1 only occurs in 6 out of 30 possible combinations of robot hazard and swarm behaviour. 15 out of the 30 hazard scenarios have no effect at all on swarm behaviour, and the remaining 9 have only minor, non-serious, effects.

3.2 Systematic errors

Systematic errors are those aspects of the design that could allow the system to exhibit undesirable behaviours. For swarm engineered systems analysis of systematic errors clearly needs to take place at two levels: in the individual agent and for the swarm as a whole. Analysis of systematic errors in the individual agent should be helped by the relative simplicity of the agents. In a previous paper [11] we apply a Lyapunov stability approach to develop a methodology for the provably stable design of the individual robots of the swarm.

For a swarm as a whole, the notion of a systematic error can only refer to an error in the *specification* of the desired swarm behaviours as they are an emergent consequence of the agent interactions, having no explicit physical embodiment or implementation. The dependability of a swarm will be compromised if there is some defect in the specification of the behaviour at the swarm level that excludes any possibility that agents can be designed to achieve the required emergent behaviour. Therefore, any rigorous swarm engineering methodology must include techniques for the specification of swarm behaviour that is logically consistent within itself such that there are no contradictory requirements, and internally complete so that there are no omissions in the specification compared to its requirement (particularly with respect to environmental conditions). In [24] we have recently explored the use of a temporal logic formalism to formally specify, and possibly also prove, the emergent behaviours of a robotic swarm. That paper showed that a linear time temporal logic formalism [9] can be applied to the specification of swarm robotic systems, because of its ability to model concurrent processes, and applied the temporal logic schema to the same wireless connected swarm case study used in the present paper, starting with the specification of individual robots and successfully building up to the overall swarm.

4 Can Swarm Systems be Robust?

In the swarm intelligence literature, the term ‘robustness’ has been used in a number of different ways. A swarm has been described as robust because:

1. It is a completely distributed system and therefore has no common-mode failure point;
2. it is comprised of simple and hence functionally and mechanically reliable individual robots [18];
3. it may be tolerant to noise and uncertainties in the operational environment [19];
4. it may be tolerant to the failure of one or more robots without compromising the desired overall swarm behaviours [13], and
5. it may be tolerant to individual robots who fail in such a way as to thwart the overall desired swarm behaviour.

When we speak of failure of the swarm to achieve the desired overall swarm behaviour we need to ask "failure to do what, exactly?" One of the defining characteristics of robotic swarms is that task completion is hard to pin down. There are two reasons. Firstly, because task completion is generally only in the eye of the beholder; the robots themselves often cannot know when the task is complete, either because their simplicity precludes the sensing or computational mechanisms to detect the condition of task completion, or because their limited localised sensing means they cannot see enough of the environment to be able to get the big picture (or both). In the case of object clustering, for instance [18], robots that are left to run after they have done their work may even, in time, disturb and uncluster the objects, only to then form them in a different place. Secondly, in swarm robotics, task completion is often defined by some statistical measure rather than a hard determined outcome.

We can conclude that our case study swarm does indeed merit the characterisation of 'robust', although not just because of its inherent parallelism and redundancy. Our swarm's high level of robustness is a result of several factors: parallelism of multiple robots; redundancy characterised by a sub-optimal approach to the desired overall swarm functionality (in common with the natural systems from which swarm intelligence takes its inspiration); the fully distributed approach with no 'system-wide' vulnerability to hazards; the functional simplicity of individual robots, and the swarm's unusual tolerance to failure in individual robots. It is useful to reflect on the fact that this level of fault-tolerance comes for free with the swarm intelligence paradigm, that is, without special efforts to achieve fault tolerance by the designer. Contrast this with conventional complex distributed systems that require considerable design effort in order to achieve fault tolerance.

5 Can Swarm Systems be Reliable?

In this section we explore a number of possible reliability models for a robot swarm. The purpose of a reliability model is to enable the estimation of overall system reliability, given the (known) reliability of individual components of the system [8]. Reliability R is defined as the probability that the system will operate without failure, thus the unreliability (probability of failure) of the system, $P_f = 1\text{-}R$. In our case the overall system is the robot swarm and its components are the individual robots of the swarm.

From a reliability modelling perspective a swarm of robots is clearly a parallel system of N components (robots). If the robots are independent, with equal probability of failure p, then the system probability of failure is clearly the product of robot probabilities of failure. Thus, for identical robots,

$$R = 1 - p^N \quad (1)$$

p can be estimated using a classical reliability block diagram approach on the individual sub-systems of the robot. Since the individual robot does not internally employ parallelism or redundancy then its reliability will be modelled as a series system, giving p less than the worst sub-system in the robot.

However, this simplistic modelling approach makes a serious and incorrect assumption, which is that the overall system remains fully operational if as few as one of its components remains operational. This is certainly not true of our case study wireless connected swarm. The desired emergent swarm behaviours require the interaction of multiple robots; our swarm taxis behaviour is a dramatic example: with only one robot the behaviour simply cannot emerge. It is a general characteristic of swarm robotic systems that the desired overall swarm behaviours are not manifest with just one or a very small number of robots. However, the question of how many (or few) robots are needed in order to guarantee a required emergent behaviour in a particular swarm and for a particular behaviour is often not straightforward.

5.1 A load-sharing approach

This leads us to suggest that a robot swarm should be reliability-modelled as a parallel *load-sharing* system since, in a sense, the overall workload of the swarm is shared between its members. A reliability model of a parallel load-sharing system takes the approach that if one component fails then the probability of failure of the remaining $N\text{-}1$ components increases; if a second component fails then the probability of the remaining $N\text{-}2$ failing further increases, and so on; see Lee *et al* [15]. While such a model is certainly appropriate for conventional load-sharing systems (think of a 4 engined aircraft with one failed engine, flying on its remaining 3 engines), its applicability is arguable in the case of a robot swarm. Consider our case study. The failure of one or more robots does not intrinsically increase the workload - and hence reduce the reliability - of the remaining, operational, robots. Only in the limited sense that failed robots might increase the task completion (object encapsulation) time of the remaining robots might there be an impact on reliability, in that the remaining robots are operational for a longer time. In a robot swarm that does perform work, for example sorting or manipulating physical objects, as in [17], then it may be the case that the failure of one or more robots does increase the workload on the remaining robots; in these cases the load-sharing reliability model may be applicable.

5.2 Case study: a multi-state approach

Finally let us consider a multi-state reliability modelling approach to our case study swarm. The FMEA analysis of section 3 showed that individual robots are not always either fully functioning or completely failed, but could be in one of a number of hazard states that we labelled as $H_1...H_6$. States $H_1...H_5$ correspond to partial failure states, state H_6 is completely failed.

The FMEA revealed that the most critical hazard state is H_1, giving rise to swarm failure effect E_1: robot(s) with motor failure 'anchoring' the swarm (table 3). Thus, from a reliability point-of-view let us make the simplifying assumption that robots are in one of three states: fully operational, state H_1 or state H_6 completely failed.

If the probability of failure of a robot in state H_1, $p_1 = P(H_1)$ and the probability of failure in state H_6 is $p_6 = P(H_6)$, then clearly the reliability of one robot $r = 1-p_1-p_6$. For N robots in the swarm, the reliability of the swarm could be modelled as,

$$R = (1-p_1)^N - p_6^N \quad (2)$$

In fact plotting equation 2 for a range of values of N interestingly gives us an optimum value for swarm size in order to maximise the swarm reliability. It is trivial to find the optimum N for given values of p_1 and p_6 by taking the derivative of equation 2 with respect to N and equating to 0 (see [8]). Clearly we expect $p_1 << p_6$, but this analysis does give surprisingly low 'optimum' values for swarm size N. For example if $p_6 = 0.1$ (which is rather unreliable) and $p_1 = 0.001$ we find the optimum swarm size is between 3 and 4 robots.

Although this may appear to be a meaningless result, it is not. It tells us, firstly, that with the rather larger swarm sizes that we need in order to bring about the desired emergent swarm behaviours we are operating with a sub-optimal swarm size in terms of reliability. Secondly, and perhaps more importantly, this analysis strengthens the conclusion of the FMEA of section 3, that we need to endeavour to minimise, or ameliorate, the likelihood of hazard H_1.

6 Discussion and Conclusions

Many distributed systems in common use today are, in one sense or another, safety critical. Power distribution networks, telecommunications networks and air traffic control systems are all complex distributed systems, the failure of which may, directly or indirectly, endanger life. Although these systems are distributed they typically rely upon hierarchical command and control architectures which, with increasing complexity, are increasingly difficult to design, implement and test and, arguably, impossible to 100% verify and validate; the Denver Airport baggage handling system failure [6,10] is a good example of this problem.

Although it may take a leap of imagination[2] to consider (say) an air traffic control system based upon the principles of swarm intelligence, it might be that very large scale future systems can only be achieved with such an approach. Current centralised architectures may already have reached the limit of verifiable scalability. For these reasons we argue that it is timely and appropriate to ask whether systems based upon the swarm intelligence paradigm might be suitable for application to safety-critical distributed systems. This paper has explored that question by considering swarm intelligent systems from dependability, robustness and reliability perspectives.

Using a wireless connected robot swarm for an abstract 'hazard containment' task as a case study we have assessed the dependability of a robot swarm using FMEA and reliability modelling approaches.

The FMEA case study showed that our robot swarm is remarkably tolerant to the complete failure of robot(s) but - perhaps counter-intuitively - is less tolerant to partially failed robots. For the swarm of our case study a robot with failed motors, but all other sub-systems functioning, can have the effect of anchoring the swarm and hindering or preventing swarm motion (taxis toward the target). This leads us to two conclusions (1) analysis of fault tolerance in swarms critically needs to consider the consequence of partial robot failures, and (2) future safety-critical swarms would need designed-in measures to counter the effect of such partial failures. For example, we could envisage a new robot behaviour that identifies neighbours who have partial failure, then 'isolates' those robots from the rest of the swarm: a kind of built-in auto-immune response to failed robots.

This paper's study of reliability models shows that a multi-state reliability model is needed in order to account for the partially failed robots identified by FMEA. We have shown that a multi-state reliability model can have interesting implications for optimum swarm size (from a reliability perspective). Further work is needed to study reliability models for swarm systems including, for instance, study of the multi-state k-out-of-n reliability model, in which k would be the minimum number of robots needed for acceptable overall swarm functionality [12].

At the time of writing the authors are not aware of any real-world application of embodied swarm intelligence. Thus no one has yet faced the task of formally validating a system based upon swarm intelligent design. Safety-critical systems are typically made so by means of (a) redundancy and (b) formal approaches to specification, design, implementation and test [16]. This paper has focussed on the inherent parallelism and hence redundancy and fault-tolerance of swarm systems. In related work we are developing formal approaches to provable single agent design [11], and to specification and proof of emergent behaviours in swarm systems [24]. Others within the field are developing mathematical modelling techniques [17]. We are, however, a long way from having a complete set of tools and methodologies for the engineering of dependable systems based upon the swarm intelligence paradigm [22].

This paper has argued that swarm intelligence has the potential for systems with remarkable levels of robustness, in the sense of tolerance to failure. A level of robustness that far exceeds that which can easily be achieved with conventional approaches but comes, in effect, for free (that is without special effort on the part of the designer). Not least for this reason we believe that swarm intelligent systems do merit further investigation for application to safety-critical systems.

References

[1] Anderson T, Avizienis A, Carter WC, "Dependability: Basic Concepts and Terminology", *Series: Dependable Computing and Fault-Tolerant Systems Volume 5*, Laprie, J-C (ed), Springer-Verlag, New York (1992).

[2] Although we should remember that very large flocks of birds manage remarkably well without air traffic control.

[2] Beni G, "From Swarm-Intelligence to Swarm Robotics", in: Şahin E and Spears WM (eds) *SAB'04 workshop on Swarm Robotics*, LNCS 3342, pp 1-9, (2005).
[3] Bonabeau E, Dorigo M, Théraulaz G, *Swarm Intelligence: from natural to artificial systems*, Oxford University Press (1999).
[4] Bonabeau E, Théraulaz G, "Swarm Smarts", *Scientific American*, pp 72-79, March (2000)
[5] Dailey KW, *The FMEA Handbook*, DW Publishing (2004).
[6] Donaldson AJM, "A Case Narrative of the Project Problems with the Denver Airport Baggage Handling System", *Tech. Rep. TR2002-01*, Software Forensics Centre, Univ. of Middlesex, UK 2002.
[7] Driscoll K, Hall B, Sivencrona H, Zumsteg P, "Byzantine Fault Tolerance, from Theory to Reality", *Proc. 22nd Int'l. Conf. Comp. Safety Reliability & Security* (SAFECOMP03), Edinburgh, Scotland, October 2003.
[8] Elsayed EA, *Reliability Engineering*, Addison Wesley Longman (1996).
[9] Fisher M, "Temporal Development Methods for Agent-Based Systems", *Journal of Autonomous Agents and Multi-Agent Systems*, Vol.10 No.1, pp 41-66 (2005).
[10] Gibbs WW, "Software's Chronic Crisis", *Scientific American*, pp72-81, September (1994).
[11] Harper CJ, Winfield AFT, "A Methodology for Provably Stable Behaviour-based Intelligent Control", *Robotics and Autonomous Systems*, Vol.54 No.1, pp 52-73, (2006)
[12] Huang J, Zuo MJ and Wu Y, "Generalised Multistate k-out-of-n:G Systems", *IEEE Trans. on Reliability*, Vol.48 No.1, (2000).
[13] Kazadi S, Kondo E, Cheng A, "A Robust Centralized Linear Spatial Flock", *Proc. IASTED International Conference on Robotics and Applications*, Hawaii (2004)
[14] Kubik A, "Towards a formalisation of emergence", *Artificial Life*, Vol.9, pp 41-65, (2003).
[15] Lee SJ, Durham SD, Lynch JD, "On the calculation of the reliability of a general load sharing system", *Journal of Applied Probability*, 32, pp 777-792, (1995).
[16] Leveson NG, "Chapter 19: Designing for Safety", *Safeware: System Safety & Computers*, Addison Wesley (1995).
[17] Martinoli A, Easton K, Agassounon W, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation", *Int. Journal of Robotics*, Vol.23 No.4, pp 415-436 (2004).
[18] Melhuish C, *Strategies for Collective Minimalist Mobile Robots*, Professional Engineering Publishing (2001)
[19] Mondada F, Guignard A, Colot A, Floreano D, Deneubourg J-L, Gambardella L, Nolfi S, Dorigo M, "SWARM-BOT: A new concept of Robust All-Terrain Mobile Robotic System", *Technical report, LSA2-I2S-STI*, EPFL, Lausanne (2002).
[20] Nembrini J, Winfield A, Melhuish C, "Minimalist Coherent Swarming of Wireless Connected Autonomous Mobile Robots", *Proc. Simulation of Artificial Behaviour '02*, Edinburgh, August (2002).
[21] Nembrini J, *Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots*, PhD Thesis, University of the West of England, Bristol (2004).
[22] Winfield AFT, Harper CJ and Nembrini J, "Towards Dependable Swarms and a New Discipline of Swarm Engineering", in: Şahin E and Spears WM (eds) *SAB'04 workshop on Swarm Robotics*, LNCS 3342, pp 126-142 (2005).
[23] Winfield AFT, Nembrini J, "Safety in Numbers: Fault Tolerance in Robot Swarms", *Int. J. Modelling, Identification and Control*, Vol.1 No.1, pp 30-37, (2006).
[24] Winfield AFT, Sa J, Fernández-Gago MC, Dixon C, Fisher M, "On Formal Specification of Emergent Behaviours in Swarm Robotic Systems", *Int. J. Advanced Robotic Systems*, Vol.2 No.4, pp 363-370, (2005).
[25] Yokobayashi Y, Collins CH, Leadbetter JR, Arnold FH, Weiss R, "Evolutionary Design of Genetic Circuits and Cell-Cell Communications", *Advances in Complex Systems*, 6(1), pp 37-45, (2003).