Surprise versus unsurprise: Implications of emergence in robotics

Edmund M.A. Ronald [a,b,*], Moshe Sipper [b]

[a] *Centre de Mathématiques Appliquées, Ecole Polytechnique, 91128 Palaiseau Cedex, France*
[b] *Logic Systems Laboratory, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland*

Abstract

We examine the eventual role of surprise in two domains of human endeavor: classical engineering and what we call "emergent engineering", with examples relevant to the field of robotics. Placing ourselves within the formal framework of the recently proposed "emergence test", we argue that the element of surprise, central in the test, serves to illuminate a fundamental difference between industrial and autonomous robots: unsurprise is demanded of classically engineered automation, while a mild form of surprise — unsurprising surprise — must of necessity be tolerated in biologically inspired systems, including behavior-based robotics. © 2001 Published by Elsevier Science B.V.

Keywords: Emergence test; Emergent engineering; Evolutionary robotics; Artificial neural networks

"Fascinating" is a word I use for the unexpected. "Interesting" shall suffice here.

Mr. Spock, *Star Trek* (the original TV series)

1. Introduction

One of the significant dividing lines that can be drawn within the field of robotics is that between industrial robots and autonomous ones. Industrial robots are epitomized by the robotic "arm", as well as by the automated guided vehicles (AGVs) which follow fixed trajectories and now proliferate in transporting materials in offices and production lines of anything from T-shirts to automobiles. We distinguish such devices from autonomous robots which are *expected* to demonstrate some form of autonomy, namely an ability to adapt to modest changes in their environment in ways that would be beyond them if their behavior were rigidly predefined in advance.

Where autonomous, and adaptive, robotics are concerned, investigators currently research bio-inspired techniques and this tendency will most likely move into engineering practice. At least two such methodologies inspired by biology are already in widespread engineering use: evolutionary algorithms and artificial neural networks. Newer bio-inspired algorithmic methods will also mature from the science to the engineering stage, e.g., cellular computing [9] and ant algorithms [3]. Collective strategies like flocking, predator–prey behavior, foraging, and stigmergy, as well as phenomena such as self-replication and self-assembly are also likely to ultimately find applications in the real world. The same bio-inspired trend can be observed in mobile robotics, where locomotive forms have gaits inspired by crawling insects,

* Corresponding author.
E-mail addresses: eronald@cmapx.polytechnique.fr
(E.M.A. Ronald), moshe.sipper@epfl.ch (M. Sipper).

swimming fish, trotting dogs, swinging monkeys, and even biped walking man. These exotic new creatures are progressively eroding the near monopoly once enjoyed by straightforwardly designed wheeled robot bases.

When engineers use technologies which originate in nature, they soon encounter the phenomenon of *emergence*, where a system displays novel behaviors that escape, frustrate or sometimes, serendipitously, exceed the designer's original intent. A case in point, artificial neural network (ANN) generalization: not only the mathematically founded learning algorithms (e.g., backpropagation) but also the surprisingly good *emergent* noise-immunity of classifier networks have allowed ANN technology to move into commercial optical character recognition (OCR) products.

Whether a boon or a nuisance, emergent phenomena in these novel adaptive systems seem puzzling and unavoidable. As regards their appearance in behavior-based robotics, Arkin recently observed that:

> Coordination functions ... are algorithms and hence contain no surprises and possess no magical perspective ... Why then does the ineffable quality of emergence arise when these behavior-based robotic systems are released in the world? Why can we not predict their behavior exactly? [1, p. 107]

2. Diagnosing emergence

The emergence label is all too often used to justify an unavoidable economy of exact explanation when striking behaviors are generated and observed, as remarked upon by Arkin:

> Emergence is often invoked in an almost mystical sense regarding the capabilities of behavior-based systems. Emergent behavior implies a holistic capability where the sum is considerably greater than its parts. [1, p. 105]

In trying to ascertain the common ground of the "unexpectedness" in experiments reported by researchers in the field of artificial life (Alife), we felt a need for an "emergence tag gun": a tool that would significantly advance our reasoning about the properties of systems labeled as emergent or non-emergent, while sidestepping the quibbling invariably associated with a rigid formal definition of some concept.

Thus, along with our colleague Mathieu Capcarrère, we resorted to an operant definition in the spirit of Turing's intelligence test [10].

In our original paper [7], the interested reader will find a basket of eight examples culled from the Alife literature and submitted to the judgement of the emergence test, as well as a discussion on the relationship between our emergence tag and the definitions proposed by other researchers. We have recapitulated only the clauses of the test below, in order to render this paper self-contained.

2.1. Formulating the emergence test

Our concern is limited to what Herbert Simon called the "sciences of the artificial" [8]; it is intended to diagnose emergence in systems that are engineered by man, not those that occur naturally (hence clause 1 of the test below). The test consists of three criteria — design, observation, and surprise — for conferring the emergence label.

Assume that the scientists attendant upon an Alife experiment are just two: a system designer and a system observer (both of whom can in fact be one and the same), and that the following three conditions hold:

1. *Design*. The system has been constructed by the designer by describing *local* elementary interactions between components (e.g., artificial creatures and elements of the environment) in a language $\mathcal{L}_1$.
2. *Observation*. The observer is *fully aware* of the design, but describes *global* behavior and properties of the running system, over a period of time, using a language $\mathcal{L}_2$.
3. *Surprise*. The language of design $\mathcal{L}_1$ and the language of observation $\mathcal{L}_2$ are distinct, and the causal link between the elementary interactions programmed in $\mathcal{L}_1$ and the behaviors observed in $\mathcal{L}_2$ is *non-obvious* to the observer — who therefore experiences surprise. In other words, there is a cognitive dissonance between the observer's mental image of the system's design stated in $\mathcal{L}_1$ and his contemporaneous observation of the system's behavior stated in $\mathcal{L}_2$.

   When assessing this clause of our test, one should bear in mind that as human beings we are quite easily surprised (as any novice magician will attest). The question reposes rather on how *evanescent* the surprise effect is, i.e., how easy (or

strenuous) it is for the observer to bridge the $\mathcal{L}_1$–$\mathcal{L}_2$ gap, thus reconciling his global view of the system with his awareness of the underlying elementary interactions.

The above three clauses, relating design, observation, and surprise describe our conditions for diagnosing emergence, i.e., for accepting that a system is displaying emergent behavior.

2.2. Conferring the emergence label on ANN classifiers

We will now use the emergence test to justify our conferring the emergence label on ANN classifiers. These are ANN which take the description of a pattern as input, and assign this input pattern to one of a number of predetermined classes. Handwritten-character recognizers fall into this category, outputting a character value for each input gesture.

- *Design*. The design language $\mathcal{L}_1$ is that of artificial neuron transfer-function definitions, network topologies, and synaptic weights.
- *Observation*. The observation language $\mathcal{L}_2$ is that of input–output behavior, i.e., input patterns and class-membership assignments.
- *Surprise*. While fully aware of the underlying neuronal definitions of the topological connections and of the synaptic weights, the observer nonetheless marvels at the performance of the network, in particular its ability to generalize and classify novel inputs — previously unseen patterns — a behavior which he *cannot* fully explain, or predict without exact computation.

Diagnosis. Emergent behavior is displayed by ANN classifiers.

2.3. Diagnosing emergence in evolutionary robotics

We will now use the emergence test to justify our conferring the emergence label on the learning process for robots whose behavior is evolved via evolutionary-computing techniques [4]. These robots learn behaviors such as obstacle avoidance, wandering, and wall-following, via algorithms inspired by Darwinian evolution.

- *Design*. The design language $\mathcal{L}_1$ is that of basic robot sensations and actions, including sensor readings (e.g., visual, proximity, sonar) and motor actions (e.g., wheels, grippers).
- *Observation*. The observation language $\mathcal{L}_2$ is that of evolved high-level behavior, wherein the robot acts (and perhaps interacts) within its environment (e.g., obstacle avoidance, wandering, wall-following, flocking, and group foraging). Note that the fitness function of the evolutionary algorithm is usually specified in $\mathcal{L}_2$ terms.
- *Surprise*. While fully aware of the underlying behavioral building blocks, the observer nonetheless marvels at the performance of the evolved robot(s).

Diagnosis. Emergent behavior is displayed by evolved or evolving robots. (In several recent works, researchers have added yet another emergent technique on top of the evolutionary process — ANN [6] — which serves to further widen the $\mathcal{L}_1$–$\mathcal{L}_2$ gap.)

Thus, by conferring the *emergence label* on evolutionary robotics, we formally acknowledge that a significant degree of surprise accompanies any thoughtful consideration of the evolution of high-level behavior. At the present state of our practice of the art of robotics, the surprise cited centres on the evolutionary learning process itself, and in many cases will carry over to the actual behaviors finally evolved.

3. Categories of surprise

In his book *Scientific Literacy and the Myth of the Scientific Method*, Henry H. Bauer wrote [2]:

To make sense of the tension between innovation and conservatism in science, more helpful than the banal distinction between what is known and what is not known is the discrimination of three categories: the known, the known unknown, and the unknown unknown.

In the same vein, when reflecting on the degree of surprise felt by the observer, we would say that there are three categories of surprise: (1) unsurprise (i.e., no surprise); (2) unsurprising surprise, where our surprise is confined within well-defined bounds; (3)

surprising surprise, where we are totally and utterly taken aback.

We build on this by stating that the products of classical — non-emergent — engineering and the products of emergent engineering are perceived very differently in a social context, in large part, owing to the different categories of surprise involved: unsurprise in classical engineering and unsurprising surprise in emergent engineering (we discuss the third category — surprising surprise — and its manifestation in the field of artificial life in [6]).

4. Classical engineering: unsurprise

The mail-cart scenario. Your company, which distributes AGVs, has won the tender to automate mail distribution in a large office building. The AGVs which you will use employ painted-line guidance [5], in that they follow predefined paths marked on the floor by means of a fluorescent die. Your boss has assigned you the task of laying down the guidance tracks. The specifications you are given are extremely vague, as you have to tackle the existing office and cubicle layout and route tracks through it which your automated cart can negotiate without getting jammed or creating a human traffic jam in tight spots.

The mail client's wishes in the observation language $\mathcal{L}_2$ describe (albeit very vaguely) observable (functional and behavioral) properties of the cart as it drives itself around: the areas it should visit at least once per round, the fact that it should not get stuck, and that it should concede a certain minimal amount of *lebensraum* to humans who circulate in the same areas. Now you, the engineer, will create a design in $\mathcal{L}_1$, i.e., a track layout, ensuring that the cart during its operation meets these *behavioral* specifications.

There should be no surprises in the scenario described above: in classical engineering we always seek *unsurprise* (no surprise). The mail cart can be built and programmed to meet the specs *perfectly*, as easily proved by a walk to your local office materials handling supplier. The engineer in these non-emergent classical domains has at his disposal a set of scientific theories (in this case an understanding of geometry), and sufficient experience with these robots, such that he has confidence in his mental model of how his $\mathcal{L}_1$

constructions will behave when observed in $\mathcal{L}_2$ by the "client".

5. Emergent engineering: unsurprising surprise

Whereas in the preceding section we focused on engineering with the help of a theory, which would be labeled as non-emergent, we now wish to move our attention to the case where *emergence* strides onto the scene.

The garbage-disposer scenario. The president of the company that bought your robot track-following mail cart appreciates your engineering skills, and calls you to his office one bright morning to discuss a new project. The employees in his R&D department tend to work fervently, cluttering their office floors with scraps of paper and empty beverage cans. Impressed by the performance of the roving mail distributor, the president would like to commission what *seems to him* a similar project: designing a garbage-disposal robot, which will roam about the R&D office rooms at night — and clean them up.

To solve the assigned garbage-disposal task you decide to use an emergent technology — evolutionary robotics — whose emergent behavior we discussed earlier.

The garbage-disposer scenario (*Continued*). Having built and evolved the garbage-disposal robot, you invite the president and his board to see the results, and proudly demonstrate the operation of the robot. You report to them that tests have shown the robot capable of cleaning 72% of the garbage items it finds on the floor. The president is very happy with this figure and reaches immediately for the official company checkbook. You do not of course tell the president how amazed you are at the success of the project. You sip the champagne and gaze with fascination at the strange trajectories of the evolved robot as it pursues and captures the scattered garbage items.

Your lingering sense of amazement stems from your inability to bridge the $\mathcal{L}_1$–$\mathcal{L}_2$ gap. You have no idea whatsoever how to *fully* explain the performance in $\mathcal{L}_1$ terms: sensors, actuators, control algorithms; at

best you can give partial descriptions and educated guesses. And if given different specifications — a new milieu or a different disposal policy — all you can do is re-evolve the robot on location, and evolution is a process whose goals are described in the $\mathcal{L}_2$ language. In all honesty, the emergent nature of evolutionary robots means that you — the designer — are surprised every time you think about them hard, *even when they are working as designed*!

The more you think of "engineering with emergence", or *emergent engineering* as we call it, the more it comes to resound oxymoronically. Emergent engineering, while inherently containing a non-evanescent element of surprise, seeks to restrict itself to what we call *unsurprising surprise*: though there is a persistent $\mathcal{L}_1$–$\mathcal{L}_2$ understanding gap, and thus the element of surprise does not fade into oblivion, we wish, as it were, to take in this surprise in our stride. Yes, the evolved robot works (surprise), but it is in some oxymoronic sense *expected* surprise: as though you were planning your own surprise birthday party.

6. Deploying emergent techniques in the real world

How can the engineer, in commercial practice, decide to deploy emergent approaches in engineering applications if the (unsurprising) surprise effect is omnipresent in the method employed?

During the design phase of a device, which embodies an aspect of emergence, two distinct modalities of unease may *potentially* arise due to emergence, and these modalities we wish to distinguish rather than conflate:

1. *Potential engineering difficulties.* The engineer's task of *creating a design* may be rendered difficult by the emergent aspects. For example, evolutionary computation is still a somewhat black art rather than a perfectly predictable process.
2. *Potential behavioral anomalies.* The behavior of the deployed application may manifest surprising aspects. Thus, neural network-based handwriting recognizers have been known to develop surprising allergies after hitting the market.

Let us consider the practical implications of these caveats which are due to emergence.

- Point (1) above, engineering difficulties caused by emergence, would impinge mainly on the engineer and remain invisible to the customer. Thus these difficulties would require the engineering personnel and management to accommodate a less systematic engineering process. For instance, when employing neural networks or evolutionary optimization, the designer will have to adapt to performing multiple runs of the corresponding stochastic algorithms. These runs may or may not converge to yield the desired quality of solution. Manual tuning of some parameters may prove necessary, and supervising the process can be both expensive in computing resources and psychologically frustrating. Sometimes, the process may fail altogether, after having consumed considerable time and money. With experience (maybe years) a process becomes more well understood, emergence subsides (or evanesces altogether), and the engineer's task becomes a more predictable routine.
- Point (2) above, the possibility of emergence of behavioral anomalies, impacts the company selling the product as a whole, and not only the design team. Surprises in the real world may — in litigious societies — generate costly product liability suits. ABS car brakes, e.g., are typical of an application which would benefit from the most advanced control algorithms, but where the associated legal risks are high (indeed ABS brakes have been known to fail dangerously after hitting the market).

To combat the unease described in point (2), we would argue that the only acceptable way to gain confidence in a product embodying emergent technologies is to specify an extremely rigorous testing regime: classical, non-emergent methods induce trust because they rest on well-understood theoretical models, and the envelopes of confidence of those models are known. Emergent methods are always pushing the envelope — yet there is none! Hence, a methodology is needed whereby the engineer can confidently represent to management (and ultimately to clients) that a design has been adequately tested.

Indeed, a practical application of the emergence test might be in the review of risks to which companies may subject their products before launch: when a product conforms to classical engineering practice, it will require a minimal amount of

due-diligence testing. However, the presence of any technology, diagnosed as emergent, should trigger a stringent review of the testing process before a design is signed off for production. At the very least, such a procedure might prevent embarrassment (as was the case of the Apple Newton PDA whose handwriting recognizer was lampooned nationwide in the Doonesbury comic strip). In some applications a hard-eyed review of testing procedures may save lives.

Because they are distinct, the two mentioned difficulties induced by emergence and the associated surprise can surface independently. An example of a situation where the engineer faces (1) and not (2) is in, say, a product palette packing problem, where each geometric solution to the packing is perfectly understandable and usable, even though the time to compute the packing solution fluctuates wildly, because the evolutionary algorithm invoked is stochastic. Conversely, an adaptive run-time load-balancing algorithm on a cluster of web servers may be perfectly implemented and effective, but may sometimes suffer such brittle degradation of performance under saturation load that its adoption would pose a commercial risk to the sites that run it.

It can be argued that the arduous testing procedure suggested for detecting emergent behavioral anomalies in (2) above would serve to move any difficulties with a system's behavior back into the engineer's realm of responsibility, transforming instances of (2) into instances of (1).

Emergence in engineering — and the ensuing reliability issues — are an unavoidable consequence of wanting "smart" machines that do things we cannot really specify, but can only say "I will recognize it when I see it".

References

[1] R.C. Arkin, Behavior-based Robotics, The MIT Press, Cambridge, MA, 1998.

[2] H.H. Bauer, Scientific Literacy and the Myth of the Scientific Method, University of Illinois Press, Urbana, IL, 1994.

[3] M. Dorigo, G. Di Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artificial Life 5 (2) (1999) 137–172.

[4] S. Nolfi, D. Floreano, Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-organizing Machines, The MIT Press, Cambridge, MA, 2000.

[5] S.K. Premt, C.B. Besant, A review of various vehicle guidance techniques that can be used by mobile robots or AGVs, in: H.J. Warnecke (Ed.), Proceedings of the Second International Conference on Automated Guided Vehicle Systems and 16th IPA Conference, IFS Publications, Kempston, UK, 1983, pp. 195–205.

[6] E.M.A. Ronald, M. Sipper, Engineering, emergent engineering, and artificial life: Unsurprise, unsurprising surprise, and surprising surprise, in: M.A. Bedau, J.S. McCaskill, N.H. Packard, S. Rasmussen (Eds.), Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life, The MIT Press, Cambridge, MA, 2000, pp. 523–528.

[7] E.M.A. Ronald, M. Sipper, M.S. Capcarrère, Design, observation, surprise! A test of emergence, Artificial Life 5 (3) (1999) 225–239.

[8] H.A. Simon, The Sciences of the Artificial, 2nd edn., The MIT Press, Cambridge, MA, 1981.

[9] M. Sipper, The emergence of cellular computing, IEEE Computer 32 (7) (1999) 18–26.

[10] A.M. Turing, Computing machinery and intelligence, Mind 59 (236) (1950) 433–460.

Edmund M.A. Ronald is an Affiliate Researcher at the Center for Applied Mathematics of the Ecole Polytechnique in Paris. His interests include the philosophical underpinnings of artificial life, communication in collective robotics, and evolutionary computation. Dr. Ronald has published over 20 research publications in these areas.

Moshe Sipper is a Senior Lecturer in the Department of Computer Science at Ben-Gurion University, Israel and a visiting Senior Researcher in the Logic Systems Laboratory at the Swiss Federal Institute of Technology in Lausanne. He is primarily interested in the application of biological principles to artificial systems, including evolutionary computation, cellular computing, bio-inspired systems, evolvable hardware, complex adaptive systems, artificial life, fuzzy logic, and artificial neural networks. Dr. Sipper has published over 90 research publications in these areas, including the book "Evolution of Parallel Cellular Machines: The Cellular Programming Approach" (Springer, Heidelberg, 1997).