Engineering, Emergent Engineering, and Artificial Life: Unsurprise, Unsurprising Surprise, and Surprising Surprise

Edmund M. A. Ronald[1], and **Moshe Sipper**[2]

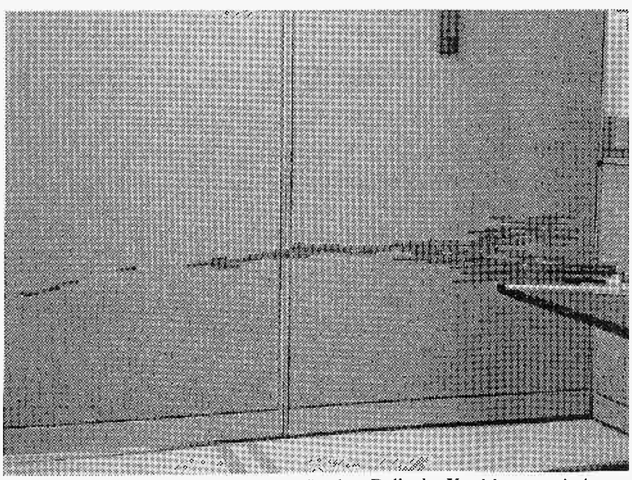
[1]Centre de Mathématiques Appliquées, Ecole Polytechnique, 91128 Palaiseau, France. eronald@cmapx.polytechnique.fr.
[2]Logic Systems Laboratory, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland. Moshe.Sipper@epfl.ch.

Abstract

We examine the eventual role of surprise in three domains of human endeavor: classical engineering, what we call "emergent engineering," and the general unrestricted field of artificial life. Our study takes place within the formal framework of the recently proposed "emergence test." We argue that the element of surprise, central in the test, serves to illuminate the fundamental differences between these three fields. This we achieve by distinguishing between three different forms of surprise: unsurprise, unsurprising surprise, and surprising surprise.

Technofright

Given the choice, would you care to entrust your one and only living body, even fleetingly, to the structurally stable bridge depicted below, which was designed by evolution [4, 5]?



We think not.

Introduction

The above bridge design appears amusing when presented as a toy, yet irrationally frightening when envisaged as a life-or-death experience. On reflection, "irrationally frightening" boils down to "what is this weird new thing?": evolutionary computation has generated an inchoate structure, totally lacking recognizable patterns such as pillars, arches, frames, stays, or cables. This bizarre creation fails altogether the test of trust-by-familiarity, a test each of us applies routinely, e.g., to food (is it healthy?), or to airplanes (are two engines really enough?). Like a new airplane design, the evolved structure evokes in us a sense of amazement, or surprise, but a rather uneasy one at that, when we think of entrusting our lives to it.

Thus, the acceptance or rejection of an engineering application is conditioned by factors which can be qualified as emotional, but rest on some quite real criteria. In particular, novelty and risk mix badly in the public eye, while widespread understanding of a method's theoretical underpinning will confirm its soundness.

Engineers commonly employ nowadays techniques classified as "emergent," and this tendency will most likely increase as artificial life (Alife) moves from the laboratory to the field. At least two bio-inspired emergent methodologies are already in widespread engineering use: evolutionary algorithms [3] and artificial neural networks [8]. Newer emergent methods will also mature from the science to the engineering stage, e.g., cellular computing [12] and ant algorithms [2].

When engineers use technologies which exhibit (putative) emergent properties, rather than classical design techniques, they are faced with a problem: how can you use an emergent technique and still guarantee the customer that the supplied product is foolproof? Cars come with guarantees; artificial neural networks should too, yet usually do not: who would sign a guarantee certificate stating that the neural network-based handwriting recognizer will always work perfectly with *your* handwriting, or stating that the voice-activated computer will reliably follow *your* commands?

A widespread sense of uneasiness undoubtedly accompanies the introduction of emergence in engineering. Although not as extreme as the jitters induced by sight of the experimental bridge reproduced above, doubts are commonly expressed about electronic devices being employed to control heavy machinery (cars, airplanes)—and which develop a will of their own. In our exploration of the roots of such misgivings we have been led to reflect on the role of surprise in engineering.

In this paper we examine three domains of human endeavor: classical engineering, what we call "emergent engineering," and artificial life. Our study takes place within the framework of the recently proposed "emergence test" [9, 10]. We argue that the element of surprise, inherent in the test, serves to illuminate the fundamental difference in the confidence we accord to products of the aforementioned three fields. We will reach a distinction between three different degrees of surprise implied in the design process: unsurprise, unsurprising surprise, and surprising surprise.

The plan of the paper is as follows: in the next section we briefly summarize the emergence test, and use it to justify conferring the emergence label on neural-network technology. Next, we present the three forms of surprise. In the subsequent three sections we discuss, respectively, classical engineering, emergent engineering, and the differences between the two. We then focus on the enterprise of artificial life and the various forms of surprise it involves, ending with some concluding remarks.

The Emergence Test

In this section we shall first recapitulate the setup associated with our emergence test, and then use the test to confer the emergence label on neural-network classifiers.

The test is an operant definition in the spirit of Turing's intelligence test, with the aim of tagging a given construction as emergent. Originally (along with our colleague Mathieu Capcarrère), we presented the *emergence test* as a sort of emergence certification mark which would garner approval from the Alife community [9, 10]. Herein, we employ the test in the manner of a definition, namely, as a tool for reasoning about the properties of emergent and non-emergent phenomena.

As noted in our previous paper [9], the test is aimed at what Herbert Simon called the "sciences of the artificial" [11], of which artificial life is a quintessential example. 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:

(i) **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 $\mathcal{L}_1$.

(ii) **Observation.** The observer is *fully aware* of the design, but describes *global* behaviors and properties of the running system, over a period of time, using a language $\mathcal{L}_2$.

(iii) **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 [9, 10].

We will now use the emergence test to justify our conferring the emergence label on artificial neural-network (ANN) classifiers. These are artificial neural networks 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.

? **Diagnosis:** emergent behavior is displayed by ANN classifiers.

Thus, by conferring the *emergence label* on the classifier networks, we formally acknowledge that a significant degree of surprise accompanies any thoughtful consideration of their behavior.

Surprise

In his book *Scientific Literacy and the Myth of the Scientific Method*, Henry H. Bauer wrote [1]: *"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, we hold that there are three categories of surprise: (1) unsurprise (i.e., no surprise); (2) unsurprising surprising, where our surprise is confined within well-defined bounds; and (3) surprising surprise, where we are totally and utterly taken aback. We shall show below that classical engineering, emergent engineering, and artificial life, are fundamentally different, in large part owing to the different categories of surprise involved.

Classical Engineering: Unsurprise

Engineering is *"the application of science and mathematics by which the properties of matter and the sources of energy in nature are made useful to people"* (Merriam-Webster online dictionary at www.m-w.com). Viewed through emergence-test spectacles, the engineer's *modus operandi* can be seen as a continual shuffling between $\mathcal{L}_1$ and $\mathcal{L}_2$. Let us demonstrate this via the following scenario:

> **The motel scenario.** Your company has won the tender to build a building in Smallville, for a well-known national chain of motels. As a civil engineer, your boss has assigned you the task of drawing up the plans. The specifications you are given consist essentially of the number of rooms of the building, the standardized motel units that must be used, and the land plot's shape (the plot is supposed large).

The motel company has spoken in the observation language $\mathcal{L}_2$, describing observable (functional and behavioral) properties: the hotel's location, its intended size, and the global expense. Now you, the engineer, will select a layout, e.g., a number of wings and the number of rooms to each wing, and draw up plans.

From $\mathcal{L}_2$ to $\mathcal{L}_1$ you need to translate the *requirement specifications* couched in $\mathcal{L}_2$ terminology into *design specifications* couched in $\mathcal{L}_1$, bricks-and-mortar terminology, i.e., actual plans for the builder. The components you will specify are prefabricated room units which are mostly trucked in, dropped into place and then hooked up. You also need to place corridors and ducting.

Your work in going from $\mathcal{L}_2$ to $\mathcal{L}_1$ is not a single drive down a one-way street: once you have an $\mathcal{L}_1$ design in hand—a layout—you will need to traverse in the opposite conceptual direction, from $\mathcal{L}_1$ to $\mathcal{L}_2$, to check that the cost requirements are met. Of course, you can see immediately whether the complete construction fits in the alloted space; in this respect the design and specification languages $\mathcal{L}_1$ and $\mathcal{L}_2$ overlap.

The $\mathcal{L}_2$ specifications of the client may be altered while the engineer's work is in progress, for instance the hotel chain may add a few rooms. Such a change might be accommodated by merely lengthening a wing. Or cost projections might cause the client to amend his initial specs. The shuffling back and forth between $\mathcal{L}_1$ and $\mathcal{L}_2$, amending things on one side and checking the effects on the other side, is at the heart of the engineering enterprise.

There are no surprises, there should be no surprises in the scenario described above: in classical engineering we always seek *unsurprise* (no surprise). The engineer in these non-emergent classical domains has at his disposal a set of scientific theories, which supply him with a satisfactory model of how his $\mathcal{L}_1$ constructions will behave when observed in $\mathcal{L}_2$ by the "client."

Regrettably though, the real-world will *always* provide some variation to the models which scientists elaborate. Concerning this, Albert Einstein said: *"As far as the laws of mathematics refer to reality, they are not certain, and as far as they are certain, they do not refer to reality."* But apart from the unavoidable gaps which will inevitably develop between the predictions of science-based models and reality, the classical engineer expects no surprises. He will expect his constructions to be used *only* within conceptual areas where the models apply. If surprises arise he will not ignore them, but rather attempt to stamp them out. Classical engineering is a conservative discipline.

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 scanner scenario.** The mayor of your town appreciates your engineering skills, and calls you to his office one bright morning to discuss a project. There have recently been several infiltrations to city hall, the mayor explains, and as a result the council has decided to install an automatic scanner at the entrance to the building. The scanner will scan the incoming crowd, and will identify potential trouble-makers, whose faces are stored in a database.

To solve the assigned face-recognition task you decide to use an emergent technology: artificial neural networks, whose emergent behavior we discussed earlier. Despite their emergent nature, artificial neural networks are used routinely nowadays by engineers. In particular, they represent an *a priori* promising choice for your scanner problem.

> **The scanner scenario (cont'd).** Having implemented the neural network-based scanner, you invite the mayor and his council to your lab and

demonstrate proudly the operation of the device. You report to them that tests have shown the network to recognize faces with a success rate of 98.5%. The mayor is very happy with this figure and reaches immediately for the official city checkbook.

On its first day of effective operation, at the entrance to city hall, the scanner attains a recognition rate of 2.3%.

You are surprised, of course, by this abysmally low performance rate and eventually figure out the $\mathcal{L}_2$ cause: a change in the illumination at city hall is to blame. But you *cannot* bridge the $\mathcal{L}_1$-$\mathcal{L}_2$ gap! You have no idea whatsoever how to explain the failure in $\mathcal{L}_1$ terms: neurons, synapses, synaptic weights. All you can do is retrain the network on location; and training is a process whose goals and procedures are fully described in the $\mathcal{L}_2$ language. But, in all honesty, the emergent nature of neural networks 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 neural network works (surprise), but it is in some oxymoronic sense *expected* surprise: as though you were planning your own surprise birthday party.

Emergent vs. Non-emergent Engineering

Before moving on to examine the wider field of artificial life and its immanent forms of surprise we ask the following: 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 can arise, and these we wish to distinguish rather than conflate:

1. The engineer's task of *creating a design* may be rendered difficult by the emergent aspects. For example, neural-network training is still a somewhat black art rather than a perfectly predictable process.

2. The behavior of the deployed application may manifest surprising aspects. Thus handwriting recognizers have been known to develop surprising allergies after hitting the market.

Despite these difficulties, emergent engineering is not impossible, as evidenced by multitudinous examples of real-world applications using neural-network techniques, and the gradual adoption of evolutionary algorithms in industry. However, we would like to consider more in detail the practical implications of these caveats which are due to emergence.

- Point (1) above, as impinging mainly on the engineer, requires essentially mental adjustment to 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. With time (maybe years) a process becomes more well understood, emergence evanesces, and the engineer's task becomes a more predictable routine.

- Point (2) above impacts the company selling the product as a whole, as surprises in the real world may generate costly product liability suits. ABS car brakes, for example, 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 such a product 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) and in some applications the review of testing may save lives.

Because they are distinct, the two mentioned difficulties induced by emergence 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 a stochastic evolutionary algorithm needs to be invoked to find it. Conversely, an adaptive 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 risk to the sites that run it.

Artificial Life: Unsurprising and Surprising Surprise

Alife is a constructive endeavor: some researchers aim at evolving patterns in a computer, some seek to elicit social behaviors in real-world robots, others wish to study life-related phenomena in a more controllable setting, while still others are interested in the synthesis of novel lifelike systems in chemical, electronic, mechanical, and other artificial media. Alife is an experimental discipline, fundamentally consisting of the observation of run-time behaviors, those complex interactions generated when populations of man-made, artificial creatures are immersed in real or simulated environments. Published work in the field usually relates the conception of a model, its instantiation into real-world or simulated objects, and the observed behavior of these objects in a collection of experiments [9].

Perusing the Alife literature one can discern three underlying motivations driving practitioners in the field. We argue that each motivational category goes hand in hand with a different category of surprise.

1. Alife as a tool for investigating natural phenomena of interest (what Langton called *life-as-we-know-it* [6]). This includes, for example, studying the relations between learning and evolution, or investigating insect behavior using robots. This view of Alife often goes by the name of "weak artificial life."

   Within this category of Alife practice we might put forward a theory and try to confirm it using Alife methods; in this case we would hope for complete and utter unsurprise: confirmation of our theory. Or, at most, we would accept facing an unsurprising surprise: some new explanation of the phenomenon in question, to emerge out of our Alife experiment, but which—though surprising—is not "too" surprising.

2. Alife as a method for tackling vaguely defined problems. When we use an evolutionary algorithm to evolve a bridge or an artificial neural network that recognizes faces we have a very precise problem definition—we are dealing with *engineering*. Using an evolutionary setup to study flocking behavior, however, involves the study of a vaguely defined problem: we are not necessarily interested in the natural phenomenon of flocking (as in item 1), nor do we have a precise definition (e.g., as embodied by a fitness function) of the exact behavior we expect.

   In this category we are using Alife as a metaphorical lunar rover to explore new, unfamiliar terrain. Though we aim to find new data or novel phenomena, we wish to stay firmly seated in our comfortable rover, that is, we are seeking unsurprising surprise. If our flocking experiment suddenly produces totally bewildering behavior (perhaps not even flocking) then we shall tumble off our virtual rover: we have just experienced surprising surprise.

3. This last category of Alife practice is usually called "strong artificial life": seeking to bring about new forms of life, or *life-as-it-could-be* as dubbed by Langton [6]. This form of Alife (as yet unattained) involves *surprising surprise*: the new form of life is, *ipso facto*, entirely novel, hence producing a strong, lingering sense of amazement.

Concluding Remarks

Surprising surprise in engineering is almost invariably a nasty surprise, from a neural network's low performance to a bridge's collapse. This kind of surprise is what we feel when we view the evolved bridge shown at the beginning of this article—hence our reluctance to traverse it. Engineering and surprising surprise do not go hand in glove. So why, when we wish to abolish surprise, would we let emergence enter engineering at all?

An insight to this paradox may be given by a trivially extreme case of the emergence test: in [9] we applied the test to Minsky's putative Society of Mind according to which mind emerges from a society of myriad, mindless components [7]. We concluded that:

> Mind is an emergent phenomenon, *par excellence*, since the observer always marvels at its appearance. [9]

Now, we are led to reflect that given a complex design task which can only be vaguely specified, *any solution to that task will be surprising* and will thus pass the emergence test!

Hence the presence of emergence in engineering may be a natural consequence of the modern trend which is leading engineering into areas where we expect machines to do things which we cannot really specify, but, like intelligence and life, can only say "I will know it when I see it!"

MISSION CONTROL: If the computer should turn out to be wrong, the situation is still not alarming. The type of obsessional error he may be guilty of is not unknown among the latest generation of HAL 9000 computers...

No one is certain of the cause of this kind of malfunctioning. It may be over-programming, but it could also be any number of reasons.

2001: A SPACE ODYSSEY

References

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

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

[3] D. B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence.* IEEE Press, Piscataway, NJ, second edition, 1999.

[4] P. Funes and J. Pollack. Computer evolution of buildable objects. In P. Husbands and I. Harvey, editors, *Proceedings of Fourth European Conference on Artificial Life*, pages 358–367. The MIT Press, Cambridge, MA, 1997.

[5] P. Funes and J. Pollack. Evolutionary body building: Adaptive physical designs for robots. *Artificial Life*, 4(4):337–357, Fall 1998.

[6] C. G. Langton. Preface. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages xiii–xviii, Redwood City, CA, 1992. Addison-Wesley.

[7] M. Minsky. *The Society of Mind.* Simon and Schuster, New York, 1986.

[8] R. Rojas. *Neural Networks: A Systematic Introduction.* Springer-Verlag, Berlin, 1996.

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

[10] E. M. A. Ronald, M. Sipper, and M. S. Capcarrère. Testing for emergence in artificial life. In D. Floreano, J.-D. Nicoud, and F. Mondada, editors, *Advances in Artificial Life: Proceedings of the 5th European Conference on Artificial Life (ECAL'99)*, volume 1674 of *Lecture Notes in Artificial Intelligence*, pages 13–20. Springer-Verlag, Heidelberg, 1999.

[11] H. A. Simon. *The Sciences of the Artificial.* The MIT Press, Cambridge, Massachusetts, second edition, 1981.

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