In Vivo Veritas: Towards the Evolution of Things

A.E. Eiben

VU University Amsterdam, The Netherlands
a.e.eiben@vu.nl

Abstract. The main thesis of the position paper is that in the near future it will be possible to create populations of animate physical objects that undergo evolution in real space and real time. The resulting systems will differ from Evolutionary Computing in two crucial aspects. First, the individuals will be physical rather than digital. This requires reproduction operators for physical objects, which forms an engineering challenge. Second, the evolutionary process will be induced by the autonomous behavior of the individuals themselves, not by some central evolutionary agency that orchestrates selection and reproduction. These differences imply severe challenges for evolutionary algorithm designers because 'tricks' that work in *in silico* may not work *in vivo*. However, overcoming these challenges will ignite the development of a new field that combines Evolutionary Computing, Robotics, Artificial Life, and Embodied AI with a great potential for engineering as well as scientific research.

Keywords: Embodied Evolution, Evolutionary Computing, Evolutionary Robotics, Self-reproducing Robots

1 Introduction

This is a position paper corresponding to the keynote I gave on the 13th International Conference on Parallel Problem Solving from Nature, a.k.a. PPSN 2014, about what I call the Evolution of Things or strongly Embodied Evolution.[1] It builds on the ideas presented in my TEDx talk (http://tinyurl.com/EibenTEDx) and the 2012 paper in the Evolutionary Intelligence journal [17]. To avoid a big overlap with these, in this paper I focus on the technical aspects from an evolutionary perspective and illuminate certain challenges and possible solutions based on earlier work of my collaborators and myself.

Perhaps the best way to introduce the underlying vision is to contrast two types of evolutionary processes we know today, programmable artificial evolution in computer models (*in silico*) and real-world natural evolution out in the wild (*in vivo*), cf. Figure 1. Note that "programmable" is meant there in a loose sense. It does not imply the ability to deterministically drive the system to some state. Rather, it means the ability to specify the details of the individuals' makeup and to prescribe rules for their behavior. The subject of this paper is the new, exciting area of research in the intersection that

[1] The term "embodied evolution" is used in [42] to describe the (distributed) evolution of controllers in a population of physical robots with fixed morphologies. To avoid confusion one could call that system "weakly embodied evolution" and use "strongly embodied evolution" for systems where the bodies evolve as well.

will feature evolvable artefacts (with designable makeup and behavioral rules) in the physical world. In other words, I will speculate about machines with evolvable bodies and minds.

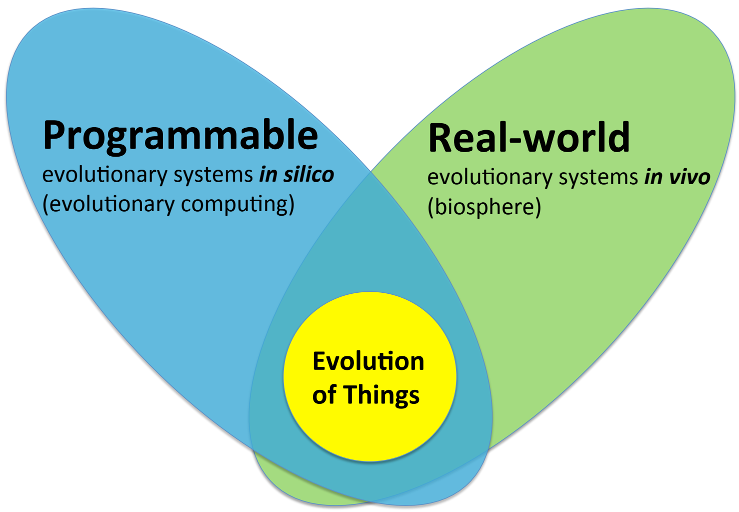

Fig. 1. The Evolution of Things takes place in real space and real time based on (self-)reproducing physical -rather than digital- entities.

Natural evolution is the force behind the emergence of Life on Earth, as established in the 19th century [9]. The invention of the computer in the 20th century made it possible to create artificial worlds and actively engineer artificial evolutionary processes in digital spaces. The resulting field, called Evolutionary Computing, was groundbreaking in that it converted evolution from a passive explanatory *theory* meant to explain a past process into an active *tool* meant to create new processes. However, an evolutionary computing process and the underlying computer models only capture the 'macro-mechanics' of evolution through a simplified genotype-phenotype mapping and an abstract selection-reproduction loop. The biochemical and physical 'micro-mechanics' are ignored and in the current practice the emphasis lies on (ab)using artificial evolution as an optimizer. Nevertheless, by the development of Evolutionary Computing and related areas in Artificial Life we –the research and user communities– have gained much experience about working with artificial evolution. We have learned to construct various forms of evolvable digital objects. We have invented and studied various selection and reproduction mechanisms, including ones that do not exist in Nature, e.g., crossover mechanisms between more than two parents [15]. And we have designed numerous evolutionary algorithms inspired by natural mechanisms, but not limited by constraints of physical or biological reality. All in all, we have developed the know-how to set up and manage artificial evolutionary processes and to use them for solving optimization, design, and modeling problems [2, 10, 18].

Evolutionary Computing is, well, computing. Producing a new individual in an evolving population is just a matter of creating a new piece of digital code. The same holds for evolving virtual creatures in Artificial Life [37]. However, as noted in [19], such systems seriously lack "the richness of matter that is a source of challenges and opportunities not yet matched in artificial algorithms". Going from digital evolutionary systems to physical ones will be a game changer in several ways and will represent a major transition from a historical perspective that brings artificial evolution closer to natural evolution [17].

2 Robots?

The Evolution of Things as I envision takes place in real space and real time, based on (self-)reproducing physical -rather than digital- entities. In order to refine this vision it is useful to distinguish 'mindless' things and 'animate' things and to state that the idea here is to evolve animate things that can sense, make decisions, and perform actions autonomously. Of course, it is possible to create a system of evolving mindless objects[2], but the case of animate objects is more interesting and promises more applications. This makes robots, a.k.a. intelligent machines, relevant because robots are physical objects that can sense, make decisions, and perform actions autonomously. One could even extend the traditional notion of robots and postulate that any kind of animate artefact or machine capable of sensing, making decisions, and performing actions autonomously is a robot, regardless of the substrate that determines its physical makeup, i.e., body, and control architecture, i.e., mind.

Carrying this view further we could say that The Evolution of (animate) Things is the same as the evolution of robots, if only we define robots in the broad sense. It could be argued that this definition is too limited because the notion of robots is not broad enough. For instance, chemists working on specially engineered molecules that can self-replicate or biologists trying to strip down living cells to make them programmable may not see their evolving entities as robots, although they could be called 'things'. Furthermore, it could be incorrect to see molecules as animate entities because do not have the ability to sense and to make decisions. However, this discussion is beyond the scope of this paper, the point I want to make is that willing to evolve animate things naturally leads to Evolutionary Robotics (ER).

Evolutionary Robotics is the combination of evolutionary computing and robotics [6, 12, 20, 33, 38, 39, 41]. ER is a field that "aims to apply evolutionary computation techniques to evolve the overall design or controllers, or both, for real and simulated autonomous robots" [39]. This approach is "useful both for investigating the design space of robotic applications and for testing scientific hypotheses of biological mechanisms and processes" [20]. The field of ER has made much progress over the last decade and a half. A recent overview, cf. [6], summarizes the key insights as follows:

[2] In such a system the objects just passively undergo evolutionary operators executed by some 'evolution manager' – quite like the digital individuals in a usual evolutionary algorithm.

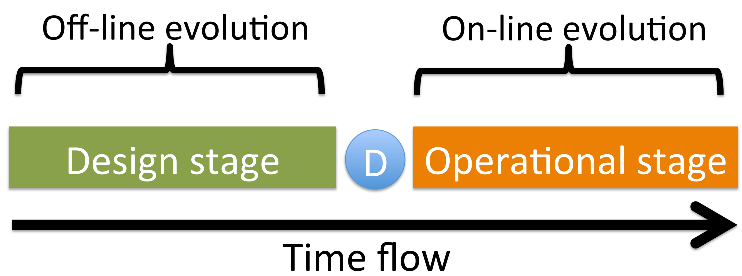

Fig. 2. Workflow of robot design distinguishing the design stage and the operational stage of the robots, separated by the moment of deployment (circle in the middle). Off-line evolution takes place in the design stage and the evolved features (usually the controllers) do not change after deployment. On-line evolution is performed during the operational period, which means that the robot's features are continually changed by the evolutionary operators.

- Manual design of a mobile robot that is autonomous and adaptive is extremely difficult.
- As an alternative, computers can 'evolve' populations of robots in a simulator...
- This evolutionary approach changes the way we view robotics: ... focus shifts to creating an evolutionary system that continuously designs and manufactures different robots with increasing abilities.

However, as noted in [6] "the use of metaheuristics [i.e., evolution] sets this subfield of robotics apart from the mainstream of robotics research" which "aims to continuously generate better behavior for a given robot, while the long-term goal of Evolutionary Robotics is to create general, robot-generating algorithms".

From the Evolutionary Computing perspective, ER is a special application area that is different from, say, combinatorial optimization. Somewhat oversimplifying, the main challenge in solving optimization problems with EAs is the ruggedness of the fitness landscape defined by the objective function. For ER applications there are two additional problems: the very weak and noisy link between controllable design details and the target feature(s) and the great variety of conditions / requirements under which a solution should prove good. For example, if we are to evolve NeuralNet controllers for a robot then the NN descriptors (direct or indirect parameters of the NN topology and weights) are the genotypes and the NN controllers form the phenotypes. Unlike in 'simple' optimization, these phenotypes cannot be directly evaluated. Rather, it is the robot behavior induced by the given controller that is observed and assessed. Thus, in usual EC we have a 3-step chain: genotype – phenotype – fitness, while in ER the chain is 4-fold: genotype – phenotype – behavior – fitness. Additionally, the behavior depends on many external factors not only on the genotype and the evaluation of a con-

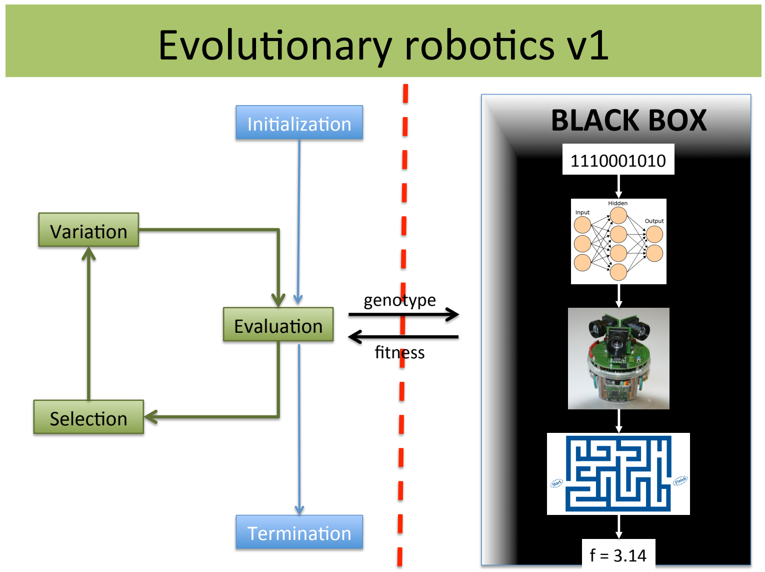

Fig. 3. Most ER applications use a quite straightforward evolutionary algorithm, only the fitness evaluations are special: these are mainly done by a simulator, sometime including occasional evaluations with the hardware in-the-loop.

troller requires running the robot for a while under different circumstances. Last but not least, desirable robot behavior is almost never defined by one single skill (except for pure research purposes). For instance, it could be required that the robot performs well in various arenas, under different light conditions regarding its skills for locomotion, collision avoidance, target following, object manipulation, and cooperation with other robots. Consequently, fitness functions in ER are inherently very noisy, very expensive, and multi-objective in terms of behavioral requirements [32].

Since this paper is not meant to be a survey of evolutionary robotics in the following I only consider two features that most ER applications share: 1) the off-line character of the evolutionary process and 2) the use of simulators. From this perspective, the Evolution of Things is beyond conventional evolutionary robotics, because it is based on on-line evolution in the real world.

2.1 Evolutionary Robotics Version 1 – Off-line Evolution

To illuminate the on-line vs. off-line aspect consider Figure 2. The usual approach in evolutionary robotics employs an evolutionary algorithm to find a good controller before the operational period of the robot. The evolutionary algorithm (EA) is quite straightforward, only the fitness evaluations are special: these are done by a simulator, possibly under different starting conditions, cf. Figure 3. When the user is satisfied with the evolved controller, then it is deployed (installed on the physical robot) and the operational stage can start. In general, the evolved controllers do not change after deployment during the operational stage, or at least not by evolutionary operators. Naturally, there are studies that use the ‘hardware in the loop’ for (some of the) fitness evaluations, but this does not change the general workflow illustrated by Figure 3, most importantly, it does not require different kinds of EAs. Let me note that this workflow also applies for

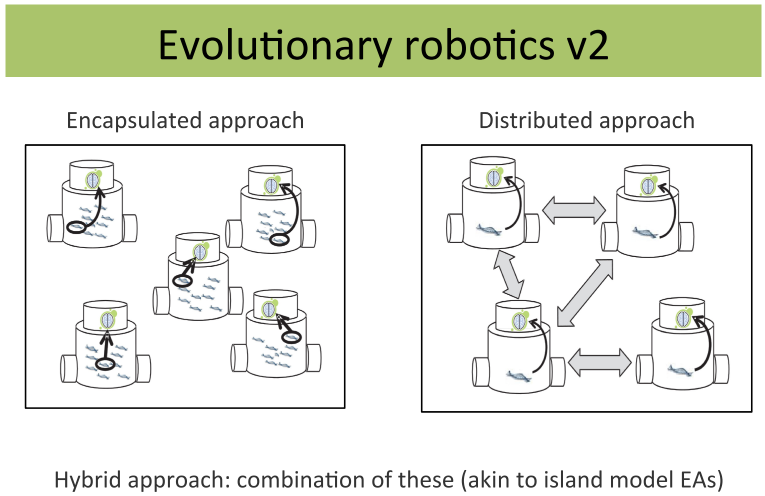

Fig. 4. In on-line ER applications the evolutionary algorithm runs in real-time on the robots themselves. Thin black arrow from the 'DNA' to the 'brain': the genotype is expressed and the corresponding phenotype (controller) is activated. Fat grey arrow between robots: interaction for mating (recombining genotypes). Fitness evaluations are done on the real hardware and they cannot be repeated for good statistics under the same conditions.

most studies that use off-line evolution for evolving morphologies. The huge majority of work in ER falls in this category belonging to the upper half of the table shown in Figure 6.

2.2 Evolutionary Robotics Version 2 – On-line Evolution of Controllers

In principle, there is an option to apply on-line evolution to evolve robots controllers during the operational period [16]. This implies that evolutionary operators can change the robots' control software even after deployment. Although this option has already been investigated early on in the history of the field, see for example [35, 42], relatively little effort has been devoted to this type of systems. This preference is not surprising, because it is fully in line with the widespread usage of EAs as optimizers, which fits the off-line approach very well. However, natural evolution is not a function optimizer, nor are evolutionary algorithms [11]. The natural role of evolution is that of permanent adaptation and using artificial evolution in this role in a group of robots requires adjustments to the usual EA setup as illustrated in Figure 4. This role is expected to become more and more important in the future of robotics. The advantages of such systems is phrased in [32] as follows:

> "Advanced autonomous robots may someday be required to negotiate environments and situations that their designers had not anticipated. The future designers of these robots may not have adequate expertise to provide appropriate control algorithms in the case that an unforeseen situation is encountered in a remote environment in which a robot cannot be accessed. It is not always

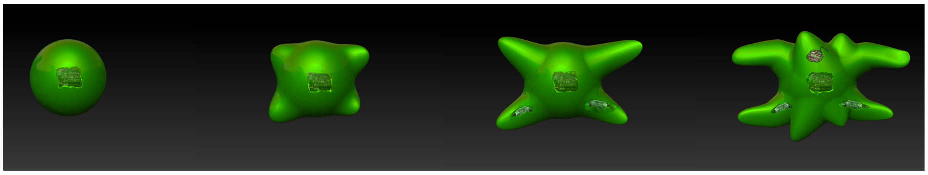

Fig. 5. Artist impression of soft robots with evolving morphologies (and corresponding controllers). Consecutive images illustrate members of different generations from the start (left) to the end (right). Courtesy of Pablo Gil-Cornaro and Claudio Rossi.

> practical or even possible to define every aspect of an autonomous robot's environment, or to give a tractable dynamical systems-level description of the task the robot is to perform. The robot must have the ability to learn control without human supervision."

From the perspective of an evolutionary algorithm designer such systems are interesting and challenging, because they have a dynamics different from usual EAs and version 1 of ER (ER–v1). For instance, such systems have two kinds of units, the robotic units and the evolutionary units. The robotic units are the physical, pre-engineered, and fixed bodies that contain the computers that run the evolutionary algorithm. However, unlike in ER–v1, these computers can move, interact, and their movements and interactions depend on the evolving controllers. The evolutionary units are the controllers that form the evolving population, they undergo selection and reproduction. These units are digital, flexible, and continually changing. The interactions between these two types of entities has not ben studied yet. In terms of the nomenclature introduced in Section1 this version of ER can be called weakly embodied. One important feature of such systems is that the number of bodies, that is robots, is given and cannot be extended. This implies 'no-go-areas' in the search space because a bad guess (a poor controller resulting from an unlucky variation operator) can be 'lethal' for the hosting robot body if tested in real hardware. While in usual EC bad guesses are just wasting time, in weakly embodied ER they can waste the robots.

2.3 Evolutionary Robotics Version 3 – On-line Evolution of Morphologies (and Corresponding Controllers)

Work concerning the evolution of morphologies is scarce and either not on-line or not physical. That is, existing work is done either in an off-line manner in computer simulations only constructing the evolved robots afterwards, see for instance [30], or in an on-line fashion but in simulation. Papers in this latter category are often positioned within Artificial Life, investigating the evolution of 'virtual creatures' [37] or 'machines' in general [3] , rather than robots in particular. As the question mark in Figure 6 indicates on-line evolution of robot morphologies and the corresponding controllers has not been done yet. The reason is quite obvious: reproduction operators for physical artefacts are much harder to implement than for digital objects. In evolutionary computing there

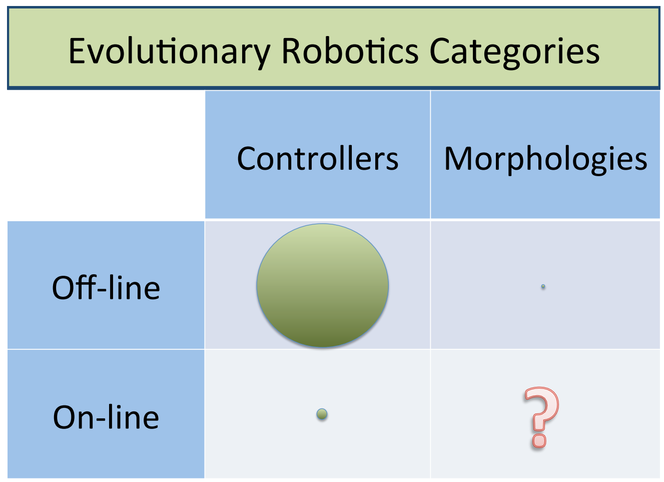

Fig. 6. Four categories within Evolutionary Robotics based on what is being evolved (controllers or morphologies) and how it is evolved (off-line or on-line). The sizes of the circles indicate the number of papers in each category. NB Circles are not at real scale.

are several mutation and crossover operators for all kinds of genotypes from simple bit-strings to complex decision trees and the construction of the resulting child(ren) is trivial in software. However, doing the same in real hardware is a different story and self-reproducing robots form one of the Grand Challenges for Evolutionary Robotics proposed in [13].

After this brief overview, the Evolution of Things can be described from an ER perspective: it amounts to strongly embodied evolution, that is, ER of the third type, where morphologies and corresponding controllers evolve on-line in the real world.

3 The Evolution of Things: Why

There are several reasons to be interested in the Evolution of Things [17]. The technology of evolvable robots offers possible applications in the future, where adapting the robot design and/or producing new robots during the operational period without human intervention is important. This can be the case in inaccessible environments, for example, colonies of mining robots that work in extreme depths under the surface of the Earth for extended periods, planetary missions, deep see explorations, or medical nano-robots acting as ‘personal virus scanners’ inside the human body. Additionally, self-reproducing robots can be evolved with the human in the loop very much like breeding livestock. This can combine the human guidance (user selection) with the creative exploratory power of evolution as used today in *in silico* evolutionary design [4, 5]. There are also benefits for scientific investigations including biological research where robots can be used as the substrate to create physical, rather than digital, models of biological systems and to study biological phenomena [21, 31, 40]. Furthermore, this new technology offers unprecedented opportunities for embodied Artificial Intelligence. In an evolving population of self-reproducing robots minds and bodies can co-evolve in the real world. This eliminates the restriction of working with fixed morphologies and opens the possibility to studying the mind-body problem in a new way [1, 7, 26, 27]. One could say that with the new technology we cannot only study how the body shapes the mind, but also how the mind shapes the body [34].

To illustrate the limitations of studying virtual creatures let me consider an investigation into the "Effects of Evolutionary and Lifetime Learning on Minds and Bodies in an Artificial Society" published in [8]. The study concerns a simulated artificial environment with a population of individuals that have a body as well as a mind. That is, some of their features effect their physical properties, like speed and strength, while other features influence their mental preferences in interacting with the environment and other agents. The paper compares two approaches to adapting these individuals. In the first approach the bodies and the minds develop through evolution, i.e., body features as well as mind features are inheritable, hence evolvable. In the second approach only the bodies evolve and the minds are adapted by lifetime-learning. In both cases the system is purely environment driven without a user-defined quantitative fitness measure. The results indicate that the first approach is able to sustain larger and more stable agent populations and maintain a higher degree of individual success. Furthermore, quite unexpectedly, the two systems differ a lot concerning the kind of bodies that emerge over time. That is, the individuals' bodies in the last populations reside in completely different segments of the physical feature space under the two regimes even though the environment is the same. This is an interesting outcome, because it means that all other things being equal, the method used for mental development has a strong effect on the development of the physical features.

Unfortunately, it is hard to establish the general relevance of this result which could just be rooted in the properties of the overly simple model of the world, the body features, and the interactions between them. In fact, the system can be physically implausible and violate some laws of physics. The simulated world may differ from the physical one to such an extent that the experimental findings are the opposite of the real world effect. Conducting this or a similar study *in vivo*, in an evolving population of real robots would eliminate these concerns. Phrasing it from a robotics perspective, in a strongly embodied evolutionary system there will be no reality gap anymore.

4 The Evolution of Things: How

To provide the algorithmic underpinning of evolving robots in real-time and real-space a conceptual framework, dubbed the Triangle of Life (ToL), has been proposed recently [14]. The ToL scheme shown in Figure 7 does not make assumptions on the physical substrate of the evolving organisms; these can be (modular) mechatronic robots, soft robots, artefacts with nonconventional bodies and forms of control, even (bio)chemical entities.[3] Therefore the ToL does not contain general recipes for the birth / morphogen-

[3] The paper [14] illustrated the components of this framework one by one using the modular robots of the Symbrion project. However, Symbrion was not aiming at physically evolving morphologies and the components of the ToL have not been integrated.

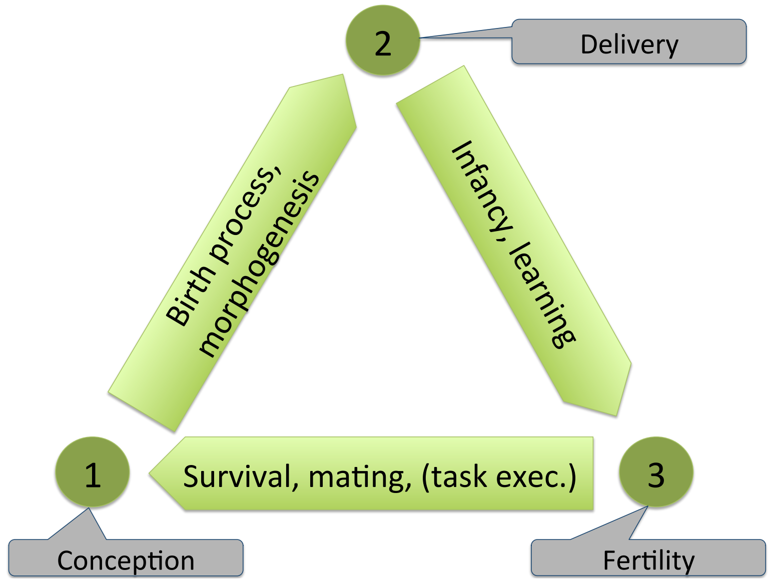

Fig. 7. The Triangle of Life after [14]. The pivotal moments that span the triangle are: 1) Conception: A new genome is activated, construction of a new organism starts. 2) Delivery: Construction of the new organism is completed. 3) Fertility: The organism becomes ready to conceive offspring.

esis operator shown by the left arrow in Figure 7. How this operator is implemented depends on the given substrate for the robot bodies. With an evolutionary computing analogy the ToL can be perceived as the equivalent of the general evolutionary algorithm loop that captures the main components of one evolutionary cycle without specifying which representation is being used, cf. Fig. 2.2. in [18].

The proverbial Cycle of Life revolves around birth. The ToL framework adopts this stance and defines a life cycle that does not run from birth to death, but from conception (being conceived) to conception (conceiving one or more children). The main idea is generic, the only significant assumption is the genotype-phenotype dichotomy. That is, it is presumed that the evolvable objects as observed ‘in the wild’ are the phenotypes encoded by their genotypes. In other words, the artefacts in question can be seen as the expression of a piece of code called the genome. As part of this assumption it is postulated that reproduction takes place at the genotypic level. This means that the evolutionary operators mutation and crossover are applied to the genotypes (to the code) and not to the phenotypes (to the physical artefacts). Nevertheless, creating new pieces of code by crossover and mutation must be followed by the physical production of the encoded entity by a birth or morphogenesis process. This is the most important distinguishing feature of the type of evolutionary systems that the ToL framework specifies.

Recall that The Triangle of Life framework is agnostic about the birth operator. However, it is important to note that birth should be implemented by a centralized system component, by a 'Birth Clinic' that constructs a new organism from a building plan, i.e., from the genome created by recombining/mutating the genomes of the parents. Distributed solutions ('pregnancy' or 'eggs') must be avoided in favor of a system with a single point of failure that can be used as a kill switch if the evolutionary process needs to be halted. I consider this an important issue of principle and emphasize that all physically embodied evolutionary systems of the future must be designed with a shutdown guarantee.

In my view such systems of evolving robots (in the broad sense) implemented through the ToL framework represent a new class of Artificial Life. Regarding the question how such a system might be used two contrasting applications present themselves. One as an engineering solution to a requirement for multiple robots in extreme unknown or dynamic environments in which the robots cannot be specified beforehand or have to be (re)adjusted to the changing conditions. The other application is scientific. Such artificial life systems could be used to investigate the development of embodied intelligence and new types of evolutionary processes, not so much to model biological evolution – life as it is, but instead to study life as it could be.

5 Special Algorithmic Challenges

The development of strongly embodied evolutionary systems bears special relevance for the evolutionary computing community. The scientific and technical knowledge regarding artificial evolutionary systems has been accumulated within this community over the last decades. Therefore, evolutionary computing could and should play an important role in the endeavor towards the Evolution of Things. However, the transition from digital and centralized evolutionary processes to physical and distributed evolution changes essential properties of the systems known and used in EC. This implies that evolutionary algorithms will have to be adjusted to cope with the new challenges. The resulting field could be seen as the 21st century incarnation of evolutionary computing with less emphasis on computing and more on evolutionary design, construction, and interaction with the environment. It can be expected that this field will benefit from certain algorithmic mechanisms in EC such that the wheel will not have to be reinvented. For instance, on-line evolutionary algorithms require on-line parameter setting mechanisms [28]. For some parameters, such as mutation rates or mutation step sizes, several methods are known in evolutionary computing and mechanisms for strongly embodied systems could be based on these.

In the following I discuss some problems raised by the Evolution of Things and show examples of existing work in EC that can be used to provide the first steps towards possible solutions.

Population Management Population management may not be the most obvious problem raised by strongly embodied evolution, but it is literally a matter of life and death. In evolutionary computing the populations (almost) always have a fixed size, maintained by the centralized 'manager' that orchestrates the evolutionary operators. The essence of the mechanism is that survivor selection (a.k.a. replacement) is synchronized with reproduction in such a way that adding n new individuals is only possible if n old ones are removed. Likewise, n existing individuals are never removed without adding n new ones. This is certainly not the case in natural evolution. In general, situated evolution without central orchestration will rely on local, non-synchronized decisions regarding birth and death [36]. Hence, the existing individuals can be removed without adding new ones and new ones can be added without discarding old ones first. This implies that populations can shrink or grow. In extreme cases this can lead to complete extinction or overpopulation such that the evolutionary process is halted. Related work in [29, 43] addresses this issue by introducing *autonomous selection* of would-be parents as well as individuals targeted for termination in decentralized evolutionary algorithms. The mechanism has the following main features:

- Locally available global information. In particular, statistical information about the population's fitness (e.g. average fitness, min/max fitness) is available at each individual via a gossiping protocol.
- A locally executable function that determines selection probabilities for the given individual based on its own fitness and the available global information.
- An adaptation method that is regulating the parameters of the selection mechanism in each individual on-the-fly, depending on the course of the search.

Experiments demonstrate the feasibility of a fully decentralized evolutionary algorithm in which the population size can be kept stable. It is shown that parent and survivor selection can be done without central control, completely autonomously and asynchronously by the individuals themselves, yet avoiding the risk of population explosion or implosion.

The experiments cited above are carried out in traditional EA applications aiming at optimizing a given fitness function. In [22] the issue of possibly exploding or imploding populations is investigated in a more natural setting, in an ALife system where evolving agents decide autonomously and asynchronously if/when they reproduce. This is called *natural reproduction* and it is complemented by natural selection where an agent dies if it runs out of energy. The primary focus of the paper is the effect of adding individual learning (reinforcement learning) to the evolutionary mechanism with a learnable individual preference for performing the mating action. This allows for runtime control of reproduction rates and in principle it can optimally regulate population sizes. However, this also implies the possibility of unlearning mating and this is exactly what happened in the naive versions of the system, because reproduction offers no individual benefits but it does imply costs. ("Children are expensive.") Experiments showed that behavior optimal on individual level can have catastrophic effects on population level, leading to complete extinction. The paper also demonstrated that this effect can be counteracted by introducing a specific reward for the mating action that gives positive feedback to the agents, regardless the related costs. One could argue that this trick is just a reinvention of a solution known in nature, commonly called an orgasm. The system with such a special mating reward proved to be viable, although the right level of reward remained an open research question.

Twofold Fitness The real world embedding in strongly embodied evolution mandates that the population is viable, i.e., can operate in the given environment that may be unknown beforehand and/or changing over time. In the meanwhile, most man-made systems are meant to serve a purpose, i.e., be useful for their designers/users. This implies that evolution should be employed for two purposes. Firstly, to provide a force for adaptation to the environment as it does in nature and in many artificial life implementations. This allows the evolving population to survive. Secondly, to provide a force for optimization towards the objectives set by the user as in mainstream evolutionary computing and evolutionary robotics. Recent work in [23–25] offers an algorithmic framework to combine the drives for viability and utility. The **M**ulti-**O**bjective a**N**d open-**E**nded **E**volution method (MONEE) balances evolution between environment-driven adaptation and task-driven optimization. It is based on the fact that evolutionary methods have two basic selection mechanisms and uses these in different roles: survivor selection is purely driven by the environment and parent selection is based on some user defined measure of task-performance. Experiments with large swarms of (simulated) e-pucks prove that MONEE does indeed promote task-driven behavior without compromising environmental adaptation. Furthermore, it is shown that an additional market mechanism can ensure equitable distribution of effort over multiple tasks.

6 In Vivo Veritas

The central thesis of this paper is that the Evolution of Things combines the controllability and programmability of artificial evolutionary systems as used in evolutionary computing and the physical embedding of natural evolutionary systems as seen in the biosphere. The corresponding research area will be concerned with populations of animate physical objects that undergo evolution in real space and real time driven by the environment, user preferences (if applicable), and their own decisions. Thus, as noted in Section 2, such artefacts can be perceived as robots in the broad sense with evolvable minds and bodies.

The emerging field can be seen as a synergetic combination of Evolutionary Computing, Robotics, Artificial Life, and Embodied AI. It will bring great new opportunities and imply great new challenges. Certainly, the EC community knows much about how to design, use, and analyze artificial evolutionary processes, but the whole body of work on ‘taming evolution’ in computer simulations may prove just a frivolous exercise. The examples reviewed in the previous section suggest that some of the existing EC techniques could be useful in the new setting, but in fact it is impossible to verify this without trying them. Real (world) problems will call for real (world) solutions.

To consider another angle let us recall the biological relevance discussed in Section 3. From this perspective strongly embodied evolutionary systems represent physical, rather than computational, models of evolution and this makes them more suited for biologically motivated studies. Such systems may not be based on the same biochemical micro-mechanisms as the carbon-based life on Earth, but they use the same macro-mechanisms (selection and reproduction with heredity) and *they are physically plausible*. This means that experimental findings will reveal much more about the real world than pure computer simulations.

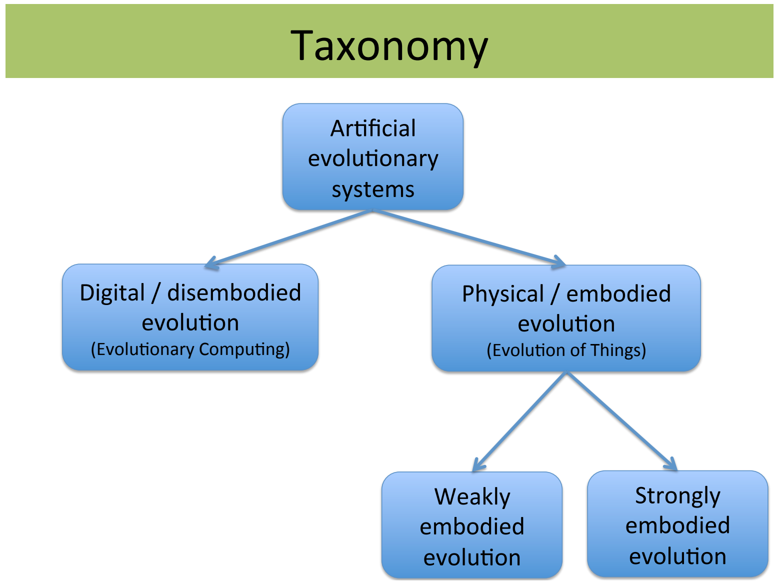

Fig. 8. Taxonomy of artificial evolutionary systems

7 Concluding Remarks

In this paper I argue that the science and technology of artificial evolution is on the verge of a major transition: from digital to physical, from software to hardware. I believe that within a few years we will have the technology for physically reproducing artefacts. Such artefacts may be 'mindless' or 'animate' and although evolving populations of 'mindless' passive artefacts will also be a novelty, the most interesting case is that of autonomous animate artefacts capable of sensing, decision making, and performing actions on their own. Such entities –robots in the broad sense of the word, not necessarily based on a traditional mechatronic substrate– will be able to actively induce an evolutionary process 'from within' –without a central evolutionary agency– in real time and real space. This new incarnation of artificial evolution will be a complete game changer confronting the designers of evolutionary mechanisms with unprecedented challenges.

Am I saying that Evolutionary Computing as we know it is doomed to disappear? Certainly not. Employing evolutionary algorithms for solving complex optimization and design problems is here to stay. The evolutionary algorithms used in these domains will become a subcategory of the bigger class of artificial evolutionary processes, that of *disembodied* / *digital* evolution, and I believe that this subcategory will remain relevant. However, I foresee that the Next Big Thing will be the emergence of *embodied* / *physical* artificial evolutionary systems, cf. Figure 8. Weakly embodied evolution will work on fixed hardware, such as populations of smart devices and/or robots that collectively evolve their control software without changing their physical makeup. Strongly embodied evolution (The Evolution of Things) will concern systems where the physical bodies co-evolve with the controllers. This will form a challenging area where I hope for exciting developments in the years to come.

Acknowledgments I would like to thank my (former) colleagues and co-authors, and several fellow researchers abroad for the inspiring discussions that helped shape my views on this subject. In particular, I am indebted to Nicolas Bredeche, Evert Haasdijk, and Alan Winfield.

References

1. M. Anderson. Embodied cognition: A field guide. *Artificial Intelligence*, (149):91–130, 2003.
2. D. Ashlock. *Evolutionary Computation for Modeling and Optimization*. Springer, 2006.
3. J. Auerbach and J. Bongard. Environmental influence on the evolution of morphological complexity in machines. *PLOS Computational Biology*, 10(1):e1003399, 2014.
4. P. Bentley, editor. *Evolutionary Design by Computers*. Morgan Kaufmann, San Francisco, 1999.
5. P. Bentley and D. Corne. *Creative Evolutionary Systems*. Morgan Kaufmann, San Francisco, 2002.
6. J. Bongard. Evolutionary robotics. *Communications of the ACM*, 56(8):74–85, 2013.
7. R. Brooks. *Cambrian Intelligence: The Early History of the New AI*. MIT Press, 1999.
8. T. Buresch, A. E. Eiben, G. Nitschke, and M. Schut. Effects of evolutionary and lifetime learning on minds and bodies in an artifical society. In *Proceedings of the IEEE Conference on Evolutionary Computation (CEC 2005)*, pages 1448–1454. IEEE Press, 2005.
9. C. Darwin. *The Origin of Species*. John Murray, London, 1859.
10. K. De Jong. *Evolutionary Computation: A Unified Approach*. The MIT Press, 2006.
11. K. DeJong. Are genetic algorithms function optimizers? In R. Männer and B. Manderick, editors, *Proceedings of the 2nd Conference on Parallel Problem Solving from Nature*, pages 3–13. North-Holland, Amsterdam, 1992.
12. S. Doncieux, N. Bredeche, and J.-B. Mouret, editors. *New Horizons in Evolutionary Robotics*, volume 341 of *Studies in Computational Intelligence*. Springer, 2011.
13. A. Eiben. Grand challenges for evolutionary robotics. *Frontiers in Robotics and AI*, 1(4), 2014. doi: 10.3389/frobt.2014.00004.
14. A. Eiben, N. Bredeche, M. Hoogendoorn, J. Stradner, J. Timmis, A. Tyrrell, and A. Winfield. The triangle of life: Evolving robots in real-time and real-space. In P. Liò, O. Miglino, G. Nicosia, S. Nolfi, and M. Pavone, editors, *Advances In Artificial Life, ECAL 2013*, pages 1056–1063. MIT Press, 2013.
15. A. E. Eiben. Multiparent recombination in evolutionary computing. In A. Ghosh and S. Tsutsui, editors, *Advances in Evolutionary Computing*, Natural Computing Series, pages 175–192. Springer, 2002.
16. A. E. Eiben, E. Haasdijk, and N. Bredeche. Embodied, on-line, on-board evolution for autonomous robotics. In P. Levi and S. Kernbach, editors, *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*, chapter 5.2, pages 361–382. Springer, May 2010.
17. A. E. Eiben, S. Kernbach, and E. Haasdijk. Embodied artificial evolution – artificial evolutionary systems in the 21st century. *Evolutionary Intelligence*, 5(4):261–272, 2012.
18. A. E. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Springer, Berlin Heidelberg, 2003.
19. C. Fernando, G. Kampis, and E. Szathmáry. Evolvability of natural and artificial systems. *Procedia Computer Science*, 7:73–76, 2011.

20. D. Floreano, P. Husbands, and S. Nolfi. Evolutionary robotics. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, volume Part G.61, pages 1423–1451. Springer, 2008.
21. D. Floreano and L. Keller. Evolution of adaptive behavior in robots by means of darwinian selection. *PLOS Biology*, 8(1):e1000292, 2010.
22. A. Griffioen, S. K. Smit, and A. E. Eiben. Learning benefits evolution if sex gives pleasure. In Z. Michalewicz and B. Reynolds, editors, *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, pages 2073–2080, Hong Kong, China, 2008. IEEE Computational Intelligence Society, IEEE Press.
23. E. Haasdijk and N. Bredeche. Controlling task distribution in MONEE. In P. Liò, O. Miglino, G. Nicosia, S. Nolfi, and M. Pavone, editors, *Advances In Artificial Life, ECAL 2013*, pages 671–678. MIT Press, 2013.
24. E. Haasdijk, N. Bredeche, and A. E. Eiben. Combining environment-driven adaptation and task-driven optimisation in evolutionary robotics. *PLoS ONE*, 9(6):e98466, 2014.
25. E. Haasdijk, B. Weel, and A. Eiben. Right on the MONEE. In C. Blum et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2013)*, pages 207–214, Amsterdam, The Netherlands, 6-10 July 2013. ACM.
26. M. Hoffmann and R. Pfeifer. The implications of embodiment for behavior and cognition: animal and robotic case studies. In W. Tschacher and C. Bergomi, editors, *The Implications of Embodiment: Cognition and Communication*, pages 31–58. Imprint Academic, 2012.
27. F. Iida, R. Pfeifer, L. Steels, and Y. Kuniyoshi, editors. *Embodied Artificial Intelligence*. Number 3139 in LNAI. Springer, 2004.
28. G. Karafotias, M. Hoogendoorn, and A. E. Eiben. Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, to appear, dii: 10.1109/TEVC.2014.2308294, 2014.
29. J. Laredo, A. E. Eiben, M. van Steen, and J. J. Merelo. EvAg: a scalable peer-to-peer evolutionary algorithm. *Genetic Programming and Evolvable Machines*, 11:227–246, 2010.
30. H. Lipson and J. B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978, 2000.
31. J. Long. *Darwin's Devices: What Evolving Robots Can Teach Us About the History of Life and the Future of Technology*. Basic Books, 2012.
32. A. L. Nelson, G. J. Barlow, and L. Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345 – 370, 2009.
33. S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, 2000.
34. R. Pfeifer and J. Bongard. *How the Body Shapes the Way We Think*. MIT Press, 2006.
35. J. B. Pollack, H. Lipson, S. Ficici, P. Funes, G. Hornby, and R. A. Watson. Evolutionary techniques in physical robotics. In *ICES*, pages 175–186, 2000.
36. M. Schut, E. Haasdijk, and A. E. Eiben. What is situated evolution? In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, pages 3277–3284, Trondheim, Norway, 18-21 May 2009. IEEE Press, Piscataway, NJ.
37. K. Sims. Evolving 3D morphology and behavior by competition. *Artificial Life*, 1(4):353–372, 1994.
38. V. Trianni. *Evolutionary Swarm Robotics – Evolving Self-Organising behaviors in Groups of Autonomous Robots*, volume 108 of *Studies in Computational Intelligence*. Springer–Verlag, Berlin–Heidelberg–New York, 2008.
39. P. Vargas, E. D. Paolo, I. Harvey, and P. Husbands, editors. *The Horizons of Evolutionary Robotics*. MIT Press, 2014.
40. M. Waibel, D. Floreano, and L. Keller. A quantitative test of Hamilton's rule for the evolution of altruism. *PLOS Biology*, 9(5):e1000615, 2011.

41. L. Wang, K. Tan, and C. Chew. *Evolutionary Robotics: from Algorithms to Implementations*, volume 28 of *World Scientific Series in Robotics and Intelligent Systems*. World Scientific, 2006.
42. R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18, Apr. 2002.
43. W. Wickramasinghe, M. van Steen, and A. E. Eiben. Peer-to-peer evolutionary algorithms with adaptive autonomous selection. In D. T. *et al.*, editor, *GECCO '07: Proc of the 9th conference on Genetic and Evolutionary Computation*, pages 1460–1467. ACM Press, 2007.