Theory Biosci. (2016) 135:131–161
DOI 10.1007/s12064-016-0229-7

ORIGINAL PAPER

Defining and simulating open-ended novelty: requirements, guidelines, and challenges

Wolfgang Banzhaf[1] · Bert Baumgaertner[2] · Guillaume Beslon[3] · René Doursat[4,5] · James A. Foster[6] · Barry McMullin[7] · Vinicius Veloso de Melo[1] · Thomas Miconi[8] · Lee Spector[9] · Susan Stepney[10] · Roger White[11]

Received: 3 December 2015 / Accepted: 29 April 2016 / Published online: 19 May 2016

Abstract The open-endedness of a system is often defined as a continual production of novelty. Here we pin down this concept more fully by defining several types of novelty that a system may exhibit, classified as variation, innovation, and emergence. We then provide a meta-model for including *levels of structure* in a system's model. From there, we define an architecture suitable for building simulations of open-ended novelty-generating systems and discuss how previously proposed systems fit into this framework. We discuss the design principles applicable to those systems and close with some challenges for the community.

This article forms part of a special issue of *Theory in Biosciences* incommemoration of Olaf Breidbach.

Vinicius Veloso de Melo: On leave from Institute of Science and Technology (ICT), Federal University of São Paulo (UNIFESP), São José dos Campos, SP, Brazil.

✉ Wolfgang Banzhaf
banzhaf@mun.ca

[1] Department of Computer Science, Memorial University of Newfoundland, St. John's, NL A1B 3X5, Canada

[2] Department of Philosophy and Institute for Bioinformatics and Evolutionary Studies, University of Idaho, Moscow, ID 83844-3051, USA

[3] Université de Lyon, INSA-Lyon, INRIA Beagle, CNRS LIRIS UMR5205, Villeurbanne , France

[4] BioEmergences Lab, CNRS USR3695, Gif-sur-Yvette, France

[5] Complex Systems Institute Paris Ile-de-France (ISC-PIF), CNRS UPS3611, Paris, France

[6] Department of Biological Sciences, and Institute for Bioinformatics and Evolutionary Studies, University of Idaho, Moscow, ID 83844-3051, USA

[7] Dublin City University, Dublin, Ireland

[8] The Neurosciences Institute, La Jolla, CA 92037, USA

[9] Cognitive Science, Hampshire College, Amherst, MA 01002, USA

[10] Department of Computer Science, and York Centre for Complex Systems Analysis, University of York, York YO10 5DD, UK

[11] Department of Geography, Memorial University of Newfoundland, St. John's, NL A1B 3X5, Canada

Keywords Modelling and simulation · Open-ended evolution · Novelty · Innovation · Major transitions · Emergence

Introduction

Background

Open-endedness, roughly defined as "the ability to continuously produce novelty and/or complexity",[1] is considered a ubiquitous feature of biological, techno-social, cultural systems, and many other complex systems that develop in a self-organized manner. There is ample literature arguing that both natural and artificial systems share this feature in various domains and that examples of open-ended systems are provided by biological evolution, human languages, legal systems, economic and financial systems,

[1] A review of the definitions found in the literature is provided in "Definitions of Open-Endedness"

music and art, science and mathematics, to name only a few.

In the field of artificial life (ALife), open-ended evolution and general open-endedness are crucial concepts, whose quest and fulfillment constitute one of its "millennium prize problems" (Bedau et al. 2000). What is surprising, however, is that open-endedness has not found a sufficiently clear-cut and stringent definition but remains ill defined, despite more than 20 years of efforts to study its realization.

Motivation

Open-ended evolution seems to be one of the fundamental characteristics of life, perhaps even a defining characteristic, motivating the search for a deeper understanding of the phenomenon. But in recent years the problem has taken on an added urgency, because the open-endedness of human socioeconomic–technological systems has begun to have profound consequences on earth's ecosphere as a whole: we are now living in the Anthropocene.

We contend that open-endedness is essentially a generic property, one that can be best understood if studied not just in one domain, biology, but in all the types of systems in which it operates, including artificial ones. Such a generic formal framework would also be an important step toward a unified scientific treatment of natural-human systems.

Informally, open-endedness refers to the ability of a system to continue producing "interesting" novelty without exhaustion. In this paper, we aim to advance the study of open-endedness from both a scientific and an engineering perspective. Classically, the former tries to understand and analyze phenomena, while the latter tries to implement, modify, and manage them. However, in the context of computational sciences and particularly ALife, one may be interested in implementing the phenomenon to understand it. That is why we here specifically discuss the conditions for a suitable implementation of open-endedness.

There are several and diverse reasons to study open-endedness. These include the following:

- We want to understand *life* as it is: how do the microscopic laws of physics generate organisms? With life being a historical system forged by evolution, understanding the open-endedness property of this evolutionary process is likely to shed light on life itself.
- We want to understand the emergence of *complex organisms*, including us: what enables a seed and a set of scattered and disparate elements to develop into a complex multicellular organism? Is development open-ended?
- We want to explain *evolution in vivo*: what are the reasons for the actual path of evolution observed on Earth? In particular, do the "Major Evolutionary Transitions" (Maynard Smith and Szathmáry 1995) that occurred in the actual evolutionary history follow some specific rules that would be those of open-endedness (if any)? Or is it contradictory even to speak of "rules" of open-endedness, as "rules" tacitly exclude or constrain some otherwise "open" possibilities?
- We want to follow the emergence of *intelligence* as a continuum: how do biological organisms generate and internalize useful models of the world in which they are embedded?
- We want to understand the change and growth of human *socioeconomic systems*, including the emergence of autonomous entities such as corporations and governments.
- We want to elucidate fundamental and methodological aspects of *ALife*: can we engineer or manage "major" transitions (Maynard Smith and Szathmáry 1995) in technical, social, or biological systems by building models able to represent them—but without hard-coding these transitions explicitly?
- We want to realize the property of open-endedness in *algorithms*: can the "universality" of physical computers support the generation of artificial open-endedness? Do results from the theory of computation and complexity in computer science apply to open-endedness?
- We want to understand the essential role of *time*: what are the characteristics of the fundamental difference between the computation of an algorithm and the mathematical "calculation" of a feature of the world, for example, a novelty or a transition?
- We want to feed back into *bio-inspired engineering* the concepts of biological complexity: is it possible to design and implement creative machines?

Hence, there is a need for a sufficiently precise definition of open-endedness to ensure that those interested in this topic can agree on what they observe, or the nature of the problem(s) they are actually trying to solve. Further, requirements need to be established for open-endedness to ultimately become a measurable and quantifiable feature of a system. Artificially designed systems will help in this process of refinement as they allow for well-prepared complex systems to be observed and manipulated in arbitrary ways.

Inception

This paper is the outcome of a workshop on "Open-ended Novelty" held in the summer of 2014 at the Department of Computer Science, Memorial University of Newfoundland, Canada. The authors come from diverse backgrounds and have different interests, yet share a common vision: we believe that open-endedness and the continual creation of

novelty are key characteristics of our universe, which can be observed at different levels of multiple systems. We think this merits a close study as it might provide valuable insights into how complex systems work and can be applied. We are convinced that a better understanding of the questions relating to the open-ended production of novelty is one of the central intellectual challenges of our time.

Outline

The structure of our argument is as follows. Before presenting our own definition, we need to install the concept of open-endedness on better foundations, hence start with a review of the literature on the topic ("Definitions of open-endedness"). Then, as our elements of definition are necessarily relative to some model, we introduce a methodological discussion of models and meta-models, and describe the main features of our own model: entities, systems, levels, information, and timescales ("A meta-model for open-endedness"). This establishes a framework in which open-endedness can be defined using three classes of novelty ("Open-endedness and novelty").

This definition also exposes certain limits to open-endedness from various perspectives, including issues of computational simulation ("Limits to open-endedness"). Given these limits, we augment our model of open-ended systems with two engineering components to provide an architecture for effective simulations of open-endedness ("Simulation of open-endedness"). We discuss a number of examples from the literature in terms of our framework and associated architecture ("Illustrative examples").

Finally, we pose a number of (grand) challenges to the community ("Conclusions").

Definitions of open-endedness

"Open-ended" may refer to different properties of a system's processes. At least two major categories of open-endedness, based on two meanings of "end", can be distinguished: (1) processes that do not stop, and (2) processes that do not have a specific objective. Biological evolution clearly fulfills both criteria, but they need not always be correlated. Absence of time limit within a small set of possible outcomes (or a single outcome) is the hallmark of attractor dynamics, by which a system settles into one stable or stationary pattern, such as a limit cycle or a chaotic trajectory: while it is still running on the short timescale, it has nevertheless reached an end state. Conversely, absence of goal within a bounded duration would be the case of a stochastic system such as a roll of the dice or a bagatelle (a board game in which a ball bounces on pins): these are short lived but can basically "land anywhere".

Combining these two categories of open-endedness (as in biological evolution) creates a third category, which is likely to be the most appropriate one from our point of view: (3) processes that do not stop *and* have no specific objective. This position emphasizes the *transient* part of dynamical systems, which in many cases can be extremely protracted to the point of never reaching convergence. Much of what goes on in living systems could be characterized as an "attempt" to remain in that transient part.

A look at the literature

There is substantial scientific literature using the expression "open-endedness" (OE) or "open-ended evolution" (i.e., OE in the context of biological evolution) from which some general features are visible.

Despite the fact that the concept strongly involves reference to biological systems and evolution in the biosphere (Bedau 1999), literature on OE mainly originates from the fields of ALife and evolutionary computation. This contrasts with the idea conveyed by many papers that life is *obviously* open-ended (Bedau 1996; Sipper et al. 1997, 1998; Bedau et al. 1998; Maley 1999; Skusa and Bedau 2002; Nehaniv et al. 2006; Stepney and Hoverd 2011; Huneman 2012). Moreover, except for a few seminal contributions Bedau (1991), Ray (1992), Harvey (1992), Barricelli (1962), most of this literature is from the last 20 years. However, many classical questions in evolutionary biology can be considered close to the question of OE. Notable examples from the biological literature are the studies by Rensch (1959), Maynard Smith (1988) and Waddington (2008), which are discussions of OE in all but name.

Although OE is a central concept in many scientific articles and communications, very few authors risk a definition that goes beyond the couple of sentences of general papers in which OE is also not defined precisely, for example, Bedau et al. (2000). Some have even referred to "*truly* open-ended evolution" without clarifying what it means.

When we take a closer look at the various definitions of OE proposed in the literature, four different categories emerge: OE can be defined as (1) "perpetual production of novelty", (2) "unbounded evolution", (3) "continual production of complexity", or (4) "the essence of life".[2] We discuss these categories one by one:

[2] Note, however, that the last definition may not be exclusive. In particular, one can define open-ended evolution as an unbounded evolutionary process and simultaneously consider that open-ended evolution is a definition of life.

1. *OE as perpetual production of novelty* This is the most classical definition of OE (Rasmussen et al. 2004; Lehman and Stanley 2011; Stepney and Hoverd 2011; Ruiz-Mirazo et al. 2008; Bianco and Nolfi 2004; Baptista and Costa 2013). It mainly follows Taylor's work (1999), although some earlier references can also be found (Kaneko 1994). Taylor defines an open-ended evolutionary system as "a system in which components continue to evolve new forms continuously, rather than grinding to a halt when some sort of 'optimal' or stable position is reached". Interestingly, he adds that "open-ended evolution does not necessarily imply any sort of evolutionary progress", which clearly distinguishes this definition from the ones based on evolution of complexity (see third item below).
2. *OE as unbounded evolution* This definition of OE is due to the seminal work of Bedau (Bedau 1991; Bedau and Packard 1992; Bedau et al. 1998), illustrated in an ecological model of L-system plants (Fernández et al. 2012) where evolution is deemed "without a definite maximum fitness, hence no optimal goal or solution to reach", or in abstract heterogenous cellular automata (Mednarch et al. 2013), where it is qualified as "long-term". It is close to (1), but complemented by statistical measures to characterize long-term evolutionary dynamics and distinguishes between three classes of evolutionary systems, including a test for OE. These measures have been refined by Channon (2003), and a fourth class has been added by Skusa and Bedau (2002). A few models have successfully passed the given test (Maley 1999; Channon 2001). However, their long-term dynamics are far from the level of diversity and complexity observed in natural systems, thus raising questions about the test itself and about the equivalence of unboundedness and open-endedness of evolution (Maley 1999).
3. *OE as continual production of complexity* In the previous definitions of OE, there is no explicit qualitative characterization of the kind of novelty produced by evolution. In this third definition, only novelties that increase the complexity of the evolving entities are considered. This definition of OE as a continual production of complexity is popular (Hutton 2002; Bentley 2003; Fernando et al. 2011; Heylighen 2012; Schulman et al. 2012; Ruiz-Mirazo and Moreno 2012; Lehman and Stanley 2012). However, this definition, which implicitly requires the existence of an "arrow of complexity" in biological evolution, is rejected by some authors, who state that open-ended evolution does not imply an increase of complexity but, simply, creates the possibility for it (Taylor 1999; Ruiz-Mirazo et al. 2004; Markovitch et al. 2012). Other authors suggest that the issues of OE (considered as continual production of novelty) and of complexity increase are linked because simple agents in artificial life have less possibility of producing novelty (Standish 2003; Soros and Stanley 2014), though this is clearly not the case in biological systems, given the history of bacteria.
4. *OE as the essence of life* Although not widely accepted in the literature, the idea that open-ended evolution could be considered as a definition of life is proposed in many papers. Ray writes that he "would consider a system to be living if it is self-replicating, and capable of open-ended evolution" (Ray 1992), an idea that was further discussed by Bedau (1996). For Ruiz-Mirazo et al. (2004), a living system is considered to be an autonomous system with the property of OE brought about by a process of evolution.

All four definitions are similar to each other. In many contributions, the difference between 'novelty' and 'complexity' is not clear at all. Moreover, they all originate from considerations on biological evolution per se, whereas the phenomenon of OE is widespread in the universe, as we stated earlier. Most importantly, these definitions generally suffer from a major drawback: they are based on terms whose definition is itself challenging ('novelty', 'complexity', 'unbounded', 'life'), as attested by qualifications added to the meaning of these terms. For example, Bianco uses the expression 'major novelty' (Bianco and Nolfi 2004), while Rasmussen et al. speak of 'adaptive novelty' (Rasmussen et al. 2004). Similarly, Taylor suggests that continuous production of novelty means that "an indefinite variety of phenotypes are attainable through the evolutionary process, rather than continuous change being achieved by, e.g., cycling through a finite set of possible forms" (Taylor 1999). Thus, not all forms of novelty would lead to OE.

We, therefore, arrive at two contrasting definitions. On the one hand, there is the idea of continual (unbounded) creation of *novelty*, which appears to be necessary but not sufficient. On the other hand, there is the notion of continual (unbounded) creation of *complexity*, which appears to be sufficient, but not necessary for OE.

One of the motivations of this paper is to offer a new definition that avoids these pitfalls. To this end, we propose to keep the first idea above, that of "continual generation of novelty", but reposition it in the well-defined context of *models* and modeling. We argue that the concept of a model is fundamental to the definition and analysis of OE and open-ended systems, because the very notion of novelty cannot be understood without the requirement for a model of the observed (and changing) system. The modeling aspect is sufficiently general to capture non-

biological systems as well, and allows us to see them from the perspective of OE.

In the next section, we discuss this point further by introducing the notion of models and "meta-models". Then, we give a few basic definitions of the elements that constitute the model that we later used to define OE.

A meta-model for open-endedness

"All models are wrong, but some are useful"

George E. P. Box, 1987

Models and meta-models

Scientific models

Scientific models are descriptive models of part of the existing world.

Physical reality comprises material and energy with structure and dynamics, exhibiting patterns in structure and behavior. We perceive some of these patterns (and even non-existent pseudo-patterns), and build models of the world. The models are abstractions; we overfit and underfit, abstract and omit, err, confabulate and imagine. Our models range from implicit mental pattern recognition and expectations, through prose, informal cartoons, sketches and diagrams, to fully formal computational and mathematical models. But they are all models, and reality can be different (richer, poorer, other). In particular, our models tend to be "crisper" than reality, identifying classes and categories where there are actually spectra, and having difficulty with "borderline" structures that fall between or on the boundaries of our categories (for example, viruses are problematic if we are using a model that insists life is a binary property).

In scientific models, if reality and our model disagree unacceptably for our purposes, it is the model that is wrong. The model needs to be corrected.

We can use our models to classify, understand, explain, and predict. Things can happen in the world that are outside the model. For example, a model of traffic flow might not include the current phase of the moon—unless it explicitly considers the effect of moon luminosity on traffic. At other times, things can happen that are in the scope of the model, but the model does not capture them sufficiently well (for example, borderline entities), or things can happen that start in the model, but move outside the model (for example, the evolution of a new species of entity), requiring a modification of the model to capture the new features. Hence, our scientific understanding of the system is model dependent (for example, if we were already using a model of the system as it exists after a speciation, then the speciation event would not move outside the model).

There are many model types used to categorize the roles that models play in scientific practice, for example, phenomenological models, computational models, scale models, analogue models. For a philosophical overview of categorizing models in science, see the studies by Frigg and Hartmann (2012), Suppes (1960). The rise of computer simulations has significantly shaped our understanding of models and their relation to theory and experiments. These issues are addressed by Humphreys (2004), Winsberg (2010) and [Weisberg (2013), ch. 4]. For a discussion of developing false models as a means to making scientific progress, see the study by Wimsatt (1987).

Engineering models

Engineering models are prescriptive or normative models of a system to be built in the world, for example, a bridge.

When building a system, we construct a model of what the system is meant to do. An engineering model describes what we are trying to bring into existence, in contrast to a *post hoc* scientific model of existing reality. It is still a model, however, in that it may not capture exactly what the system does. [We are here assuming for simplicity that the underlying scientific model of the relevant domain is well understood and correctly applied. More complete accounts are available, for example Horsman et al. (2014)].

In engineering, if reality (the engineered system) and the model of the system disagree, it is reality that is wrong, for example, if the bridge collapses. Reality needs to be "debugged" to conform more closely to the model.

Computational models

Software is a special case of an engineered system, with the model being the specification and design of the software, and the engineered system being the executing code. For example, in an object-oriented system, the model could be a collection of UML diagrams. The software is the executing code that conforms to this model (coded classes corresponding to boxes in the class diagram, with behaviors following the sequence diagrams). If no engineering model pre-exists, if the model is implicit and only the code is written, then a *post hoc* scientific-style model can be inferred. Such a model will be much messier than a designed engineering model.

Computational science and ALife can involve many different intertwined models, both engineering and scientific. In computational science, software can be used as an

executable model of a real object (biological, sociological, technological, etc.). The model of interest is the scientific model built from observing the execution of the code (analogous to the scientific models built from observing real-world behaviors). Additionally, the software will generally have been designed using engineering models (for example, UML diagrams). See Fig. 1.

So computational models can have different content depending on the specific objectives: scientific or engineering. For example, if the objective is to engineer an open-ended software system, the property of OE will be part of the engineering model of the code. Alternatively, if the objective is to model the observed behavior of the simulation of a real-world open-ended system (in the context of computational science), then OE may be present in the scientific model, derived from observing emergent behavior in the execution of the code, but it need not be present in the prior engineering model of the code. Indeed, it should not be present in the engineering model, if the objective is to determine if certain low-level behaviors can exhibit certain emergent properties.

Meta-models

As discussed above, the purpose of a model is to provide an abstract language for the relevant domain concepts. In the case of a computational model, it defines the concepts to be implemented in code. The purpose of a *meta-model* is to provide the analogous language to define such models, to provide the concepts that can be used to build the model. For example, in an object-oriented system, the meta-model would contain the concepts of 'class', 'object', 'method', 'association', and so on. Similarly, a diagram describing the Krebs cycle of an organism is a model whose meta-model contains the concepts of 'reactions', 'metabolites', or 'enzymes'. See the study by Hoverd and Stepney (2011) for an example of a meta-model capturing concepts of 'energy', 'environment', and 'organism', and its instantiation in an agent-based model. Roughly speaking, the meta-model is the key to the model in the way the key to a map lists all the graphical concepts used to draw the map. In many modeling contexts, the relevant meta-models may be implicit.

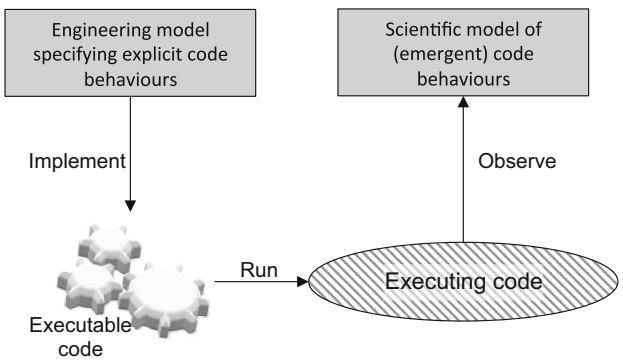

Fig. 1 Two kinds of computational model. An engineering model is used to specify the software to be implemented. A scientific model is derived from observations of the execution of the software. These models can have different structures, even about the same software

A variety of models may conform to the same meta-model. For example, the meta-model used by the model of the Krebs cycle can be used by models of other metabolic pathways. On the other hand, the same system can be captured by a variety of models conforming to different meta-models. For example, a metabolic pathway can be modeled by a set of Ordinary Differential Equations (ODEs), with a meta-model including concepts such as 'concentration' and 'enzymatic rate' (Andrews et al. 2011). Metabolic pathways can also be modeled using agent-based formalisms (Amar et al. 2008), with a meta-model including concepts such as 'agent' and 'rule'. The choice of a meta-model directly determines the limits of the models that could conform to it. In the example, the meta-model allowing ODE descriptions and resulting ODE models cannot account for spatial behavior of the metabolic pathway or for its stochastic behavior. A meta-model allowing agent-based formalisms admits models which can naturally include both behaviors.

In essence, a meta-model is a model of a model [Kleppe et al. (2003), ch. 8]. Since a model may itself be a meta-model, there is no need for a separate *concept* of a meta-meta-model, although there may be examples of such. For example, UML is the meta-model of the meta-model presented in Fig. 3. Similarly, one can view this paper as an (informal) meta-model of open-endedness.

Our entity-based multilevel meta-model

OE is a process by which continual novelty is produced in the course of a dynamic process. In this paper, we propose a scientific (descriptive) meta-model with which we illustrate our definition of OE. This particular entity-based multilevel meta-model should cover evolution in biological life, change in socioeconomic systems, and processes in ALife. Although ALife concerns artificial systems, our meta-model is not a prescriptive (normative) engineering model, but a model for describing open-ended systems, natural or artificial. Other meta-models of OE are probably possible, for example, ones supporting mathematical models formulated in terms of ODEs or PDEs. Our hope is that such a definition of OE in terms of models and meta-models will help the design of normative engineering models for implementing ALife, and provide tools to describe and understand the long-term dynamic of open-ended systems.

In our models, we are working with systems (biological, social, artificial) of entities organized into multiple levels. In the following subsections, we introduce basic definitions with respect to some (meta-)model. In a given situation, the chosen model must be a sufficiently good representation of reality to be useful.

Domain of interest

The purpose of our meta-model is to provide a language able to model any open-ended system. As explained below, this system can itself be composed of (sub-)systems. We use the term "domain of interest" to describe the open-ended system under investigation, and to avoid confusion between the system to be modeled and the systems comprising the model. Note that the concept of domain of interest is not part of our meta-model (Fig. 3), but rather a limit between what we want to study and what remains out of scrutiny. Since we do not wish to model the entire universe, the domain of interest encompasses the part of the world that we are modeling.

Entities

An *entity* is an identifiable integrated whole within the model: a "thing" with structure (organization) and behavior (activity, processing). Entities can *interact* with each other, and with their *environment*. We explicitly sidestep any discussion of the non-trivial issue of identification or demarcation of entities (Buss 1987).

Note that entities may themselves contain entities. This property naturally leads to the concept of *levels*:

- A *level-0* entity is an *atomic entity*, with no modeled internal structure.
- A *level- N* entity (where $N > 0$) is a *system entity*, having lower level entity components.

The concept of levels is refined further in "Levels".

Environment

An *environment* is a model of part of the domain of interest not modeled as explicit entities, including space, fields, flows, and so on. What is part of an environment and what is an entity is a modeling decision, and depends on what entities are chosen to be on the lowest level of our systemic hierarchy ("Levels").

A *local environment* is the part of a *system* that is not modeled explicitly as entities.

An *external environment* is that part of the domain of interest which is external to the system but exerts an influence on it. For example, it might impinge on the system via inputs (flows, forces, signals) and be in turn influenced by the system via outputs. The external environment may host other entities external to the system.

System

A *system* is a *local environment* plus a population of possibly differentiated and interacting *entities*, forming some identifiable whole, and separated from an external environment.

A system may be an *aggregate system*, comprising a collection or population of entities in a local environment, but not considered to form an entity in its own right. For example, a simple population of organisms is usually an aggregate system, and not an entity in its own right.[3]

Alternatively, a system may be a *system entity*, an entity at a higher *level* than its component self-organized entities, constraining them and their local environment via downward causation.

An *atomic entity* is not a system, as it has no internal (modeled) structure.

The specific entities comprising the system's components may change over time, for example, by recycling material. In addition, specific entities may belong to more than one system, either separated in time as entities flowing from one system to another, or simultaneously. For example, subsystems may be composed into a larger system by overlapping/intersecting some of their entities. A system needs to have a *boundary* in some space; typical such spaces include physical space, time, speed, or behavior.

Interactions

Entities *interact* with each other, and with their environment, potentially forming and dissolving systems.

Levels

We have defined a system entity to be an entity at a higher *level* than its component entities. Entities can be recursively classified into levels with corresponding interactions (Fig. 2; "Entities"). Here level-0 entities are *atomic* entities and level-$N > 0$ entities are *system entities* that contain at least two lower level entities, of which at least one is level-$N - 1$.

Following this classification of entities into levels, the domain of interest can be divided into *domain levels*: domain level N comprises all the existing/instantiated level-N entities.

[3] Though proponents of group selection, who claim that natural selection can act on populations and not just on individuals (Wilson 1997), would disagree.

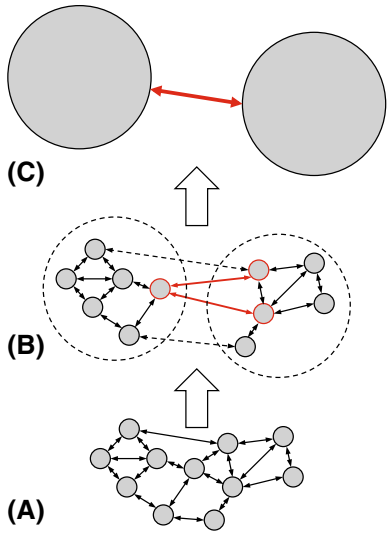

Fig. 2 Relationships between the levels of a system. **a** A collection of level-N entities (*solid circles*) interact (*thin solid arrows*) in an *aggregate system*. **b** The dynamics of these interactions stabilize the aggregates into level-$N+1$ *system entities* (*dashed circles*) and these level-$N+1$ entities contain "specialized interface elements" (*red circles*) that concentrate most of the interactions between both parts (*red arrows*). **c** These new level-$N+1$ entities can be modeled independently of their level N components and the stabilized interactions can be considered as higher level interactions. Level-$N+1$ entities may similarly aggregate and interact to form a level-$N+2$ entity, and so on

For example, if we characterize bacteria to be level-N entities, then eukaryotes could be modeled as level-$N+1$ entities, since nuclei, mitochondria, and chloroplasts all originated from ancient bacterial introgressions. Depending on modeling choices, however, these events could either be considered as a single kind of event, leading to the emergence of one higher level $N+1$, or as separate events, leading to the emergence of several higher levels (Szathmáry 2015). Colonies of bacteria that act as entities in their own right rather than mere aggregates could also be modeled as level-$N+1$ entities. Multicellular organisms could be modeled as level-$N+2$ entities, and multicellular organisms in symbiosis with populations of bacteria as level-$N+3$ entities. Alternatively, one could also think of a model where a human would be merely an environment for gut bacteria, not a separate entity on its own level. In sum, the actual levels identified depend on the model being used, which itself depends on the scientific questions being asked.

Let us clarify that we do not assume reality to be actually composed of discrete levels. Rather, levels are concepts in our meta-model. A level-$N+1$ entity is an *abstraction* of a particular collection of level-N entities that have organized and stabilized into a system. Many theories of reality include such levels, either implicitly or explicitly, for example, Koestler's "holonic hierarchy" (Koestler 1970) in which holons are defined as self-contained wholes that are at the same time-dependent parts.

The multilevel model we adopt here permits a level-N entity to have direct interactions with entities on any other level. However, having too many interactions across multiple levels is likely to render the model impossible to analyze. We suggest that, to be useful, a model of an OE system should have the overwhelming majority of interactions between levels occurring between nearest neighbors only: from N to $N+1$ and $N-1$.

Putting it all together, our resulting meta-model of environments, systems, entities, interactions, and levels is shown in Fig. 3.

A discussion of philosophical aspects of levels can be found in"On levels" in Appendix

Hierarchies and emergence in the natural world and in human systems

In the natural world, we can observe phenomena that can be modeled as many levels of systems nested within each other. The interest in level hierarchies stems from the fact that they allow us to say something about emergence of complexity in organizations (Lan 2006). More specifically, we consider higher levels of a hierarchy, as exemplified by living systems in the form of cells, tissues, organs, organisms, groups of organisms and societies, to be the result of processes of *emergence*. We shall say that the emergence of a new level in a hierarchy is a transition that produces a new system level, which happens in many cases through the composition of groups that subsequently stabilize and form entities on the higher level (see Fig. 2). It is also possible for entities to emerge by the loss of interactions, such as with simplification of gene expression pathways in biological systems. But the emergence of complexity through simplification cannot be the only pathway to emergence, since there is a bound to the number of features that can be eliminated from a system before it becomes trivial.

Niches are conditions of the environment at a certain level that influence the effects of selection on entities at that level. Niche construction is the process by which entities modify their environment, hence improving (or sometimes degrading) chances of survival (i.e., modifying selection forces) for themselves and other entities. Some argue that the process of niche construction is as important to biological evolution as is selection (Odling-Smee et al. 2003). While we do not want to engage in that discussion, we note that the active modification of the environment is a process that can be observed at several levels in the biological hierarchy as well as in social systems and other artificial hierarchical systems.

If one considers the emergence of "tentative" groups of entities to be the first (haphazard) step in the emergence of

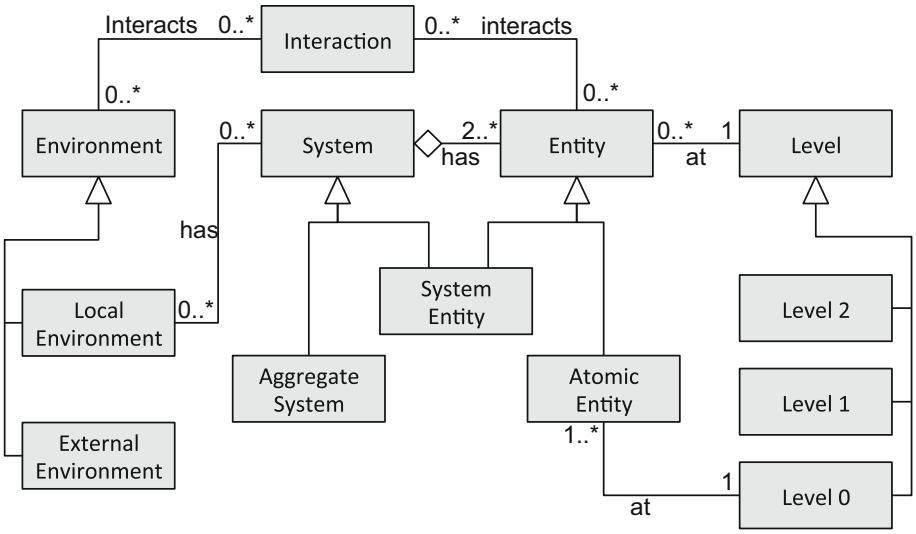

Fig. 3 Our meta-model, written in UML, defining the concepts to be used in any model conforming to it. *Boxes* are classes from which objects in the model can be instantiated. *Links* indicate associations between classes (associations being named and valued), *arrows* indicate inheritance ("kind of"), *diamonds* indicate aggregation. An `Environment` can be a `Local Environment` or `External Environment`. A `System` can be an `Aggregate System` or a `System Entity`. An `Entity` can be a `System Entity` or an `Atomic Entity`. A `System` has a `Local Environment` and two or more `Entities`. An `Entity` exists at both a system `Level` and an information structure (model) `Level`. An `Atomic Entity` exists at `Level 0`. Many `Environments` and `Entities` can have be involved in each `Interaction`

a new level, it follows that only those groups that achieve a certain degree of coherence can become candidates for entities at the emerging higher level. It is through downward causation[4] constraining the behavior of lower level entities that coherence becomes more pronounced. Competition of gradually more coherent groups of entities sets in and further channels the variations provided by the lower level, building a new system entity from patterns in the aggregate system.

We exemplify our view by discussing a specific case, Darwinian evolution, in the general model (Fig. 4). Any model of Darwinian evolution would include a population of biological individuals ("organisms") at a level N. These individuals are entities having the property of *reproduction* (sexually or asexually). Now, reproduction is governed by molecular mechanisms at the lower level ($N-1$). Since these mechanisms are error prone, level $N-1$ drives the *variation* component of Darwinian evolution. At the upper level ($N+1$) the individuals are organized into populations that interact in a finite world. These interactions result in a differentiation of the reproductive success (due to Malthusian mechanisms, predation, parasitism, altruism, etc.). This may result in many selection types, such as a neutral process (if the population is too small regarding the difference in reproductive success), natural selection in the Darwinian sense, or removal selection as in artificial systems. Ultimately, the interactions of these three levels (molecules, individuals, population) result in the realization of a Darwinian process.

In socioeconomic (or, more generally, socio-ecological) systems as well, the emergence of new entities may be identified with the emergence of downward causation, or *agency* as it is referred to in the social sciences. Individual people have agency, and are thus entities, but collections of people are often simply queues or crowds. But when their behavior in a group is formally constrained so that it is the group itself rather than the individuals composing it that exercises agency, then the formally structured group is itself a higher order entity such as a corporation, a government, or a regulatory agency; it is able to exercise causation on other entities at its level or below. As was pointed out in "Levels", there is no clean separation between levels, but in the case of socioeconomic systems the separation may be not just model dependent, but intrinsic. In Western societies, individuals are legally treated as autonomous entities. But many individuals are also "components" of higher level entities such as corporations or government agencies, and in that context must act out the appropriate role, which is not one of an autonomous individual.

The situation in human systems is even more complex because some entities in these systems, specifically governments, have acquired or given themselves the power to confer entityhood, (for example, when a business is incorporated), and the consequences can be unusual. For

[4] We circumvent the question of whether or in what sense "downward causation" exists in reality by focusing on models where we introduce it as existing.

example, if we consider the atmospheric system in isolation, then it appears to be one in which macro-scale patterns such as storms emerge, much like the flocks in boids (Reynolds 1987). Yet as previously pointed out, these do not constitute new entities and, therefore, no new, higher level is present. But if a storm satisfies the legal definition of a hurricane, then from the point of view of a governmental entity the storm itself becomes an entity, with a name, and its presence in the area causes the release of reconstruction funds that would not have been available if the storm had not achieved the status of an entity. The storm itself as a physical phenomenon may cause changes in coastal land forms, but the storm as a legal entity can induce other changes in the local geomorphology, such as dykes and drainage canals built with the reconstruction funds. Thus, depending on the point of view, the storm both is and is not an emergent higher level entity, and these points of view are inherent in the system itself.

Information

Life provides the dominant template as well as the language for thinking about OE. A key marker of the transition from non-living to living systems is that, unlike non-living systems, living systems are information dependent. They are dependent on information because they include as an essential component explicit models of themselves. Indeed, life may be defined as an entity that has a model of itself and its relations with its environment (Rosen 1991). The genome may be considered to be analogous to the code describing the model, while the phenotype in a sense executes it to survive and reproduce.

With more complex organisms such as vertebrates, the models are not just coded genomically, but also neurally based, and in human systems some are coded liguistically in oral traditions and even in external devices such as books, pictures, and software. The fact that all living systems, including economic and social systems (and the technology produced by these), incorporate models in an essential way means that they experience an additional source of indeterminacy. This indeterminacy arises from the fact that the models are necessarily incomplete or imperfect, and their imperfections are various, depending on the particular circumstances under which the models were generated, while behavior of these systems depends to a greater or lesser extent on the various models held by the individuals making up the system. This form of indeterminism associated with "knowledge creation" (i.e., model creation) was the crux of Popper's argument for an "open universe" (Popper 1982). One manifestation of this indeterminacy is the ability of agents to mislead or cheat other agents—or themselves.

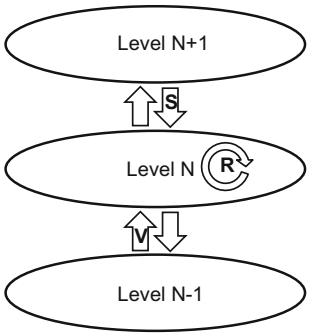

Fig. 4 A general model of three levels of an evolutionary system interacting within and between levels. Level N is the evolving layer: it is composed of entities having a (self-)replication property ("R"). Selection ("S") is a constraint imposed by the system at level $N+1$ (the "ecosystem", which may include level-$N+1$ entities such as a public health agency) upon level N. Variation ("V") is a consequence of the laws of level $N-1$ (molecules, including DNA, in Darwinian evolution) on the behavior of level N. This general model encompasses both the scientific model of Darwinian evolution, and engineering and simulation models (though these may include "shortcuts" implemented at any of the three levels; see below)

Furthermore, because the models are created in part on the basis of perceptions of the surrounding environment or system, living systems have evolved to a degree on the basis of their own perceptions of themselves. Perception must, therefore, be considered a fundamental element in evolution and OE.[5] Perceptions and misperceptions, and the model errors engendered by misperceptions, may, in some biological systems, rival genome-level phenomena as a source of the variations required for evolution; in the evolution of social systems they are clearly of primary importance. Finally, perception always involves a point of view. Therefore, in all living systems, "point of view" is an intrinsic part of the phenomenon we are trying to understand—not just something we have as we look at the problem of open-endedness and decide what approaches we want to take in trying to understand it.

Timescales

We adopt the following notion: one way to distinguish levels is by the *timescales* on which their dynamics can be observed.

A natural question to ask is what provides the conditions for a level in the model and how we can recognize these conditions to be fulfilled in a particular hierarchical system we want to observe? Since the systems we intend to consider are open and dynamical, the notion of time is an indispensable element of our discussion. Higher levels

[5] By "perception" we mean the ability of the system to sense (by whatever means) aspects of its environment, allowing it to act and react; it would not necessarily need to be a living system.

typically have slower dynamical timescales associated with them, as when groups of organisms forming a society (higher level) develop on a slower timescale than the organisms (lower level) themselves. The difference of dynamical timescales between levels can be many orders of magnitude, but might also be down to few orders of magnitude only, with extreme cases at the lower end of perhaps just one order of magnitude. When such a difference in dynamical timescales is relevant, higher levels need to slow down their development to exist or be perceived by an observer as something different from their elements.

In cases such as these, we might say that downward causation from level-N coordinates the level-$N-1$entities. We might also say that the emergence of a level-$N+1$ entity is due to regularities and coordination at level N. Hence, the level we focus on helps determine our explanation of the various levels. More discussion "on timescales and levels" can be found in the appendix.

Open-endedness and novelty

We use our meta-model, defining entities, systems and levels, to consider OE more closely, starting from the definition already mentioned:

> **Open-endedness** *is the continual generation of* **novelty**.

Given this definition, there are different kinds of OE depending on the kinds of novelty generated. We consider novelty in more detail here.

Once the relevant domain of interest has been modeled using the meta-model in our entity-based multi-level meta-model, and simulated, one can then use the model to interpret observations of its dynamics and to identify different kinds of novelty and observe different kinds of open-endedness. It follows that the kind of novelty/OE identified is *relative to some model and meta-model* of the system under investigation.

In this section, we define different types of novelty, then discuss the different classes of OE to which they can lead.

Types of novelty

Having defined the primary concepts of consideration in the meta-model, we now turn to dynamics. Due the their interactions, the systems and entities change over time. These changes can be observed and described in the model of the system as changes of the entities. Since level-N entities are composed of lower level entities and are components of higher level entities, a change observed at level N is generally not confined to a single level: it may be caused by changes at lower levels and may itself cause changes at adjacent levels.[6]

When focusing on a given change in the system, three cases can be distinguished, depending on whether the observed change can be described within the given model, within the given meta-model (but with a model change), or whether it requires a change to the meta-model. These three cases lead to three different types of novelty:

Type-0 novelty: variation

> *Variation: novelty* **within the model**.

Variation is a change to an instance of the model, a change to the values of a variable that conforms to the model.

Variation explores a pre-defined (modeled) state space, producing new values of existing variables as in, for example:

- changing the value of an integer-value variable: $x = 4$, then $x = 5$
- changing a gene to a different allele
- increasing or decreasing the size of a finch beak
- changing the number of individuals in a population (including to zero: extinction)
- swapping out one entity for another of the same type
- flipping a bit in GA with fixed length bitstrings
- changing prices of existing economic goods

In our meta-model, a variation does not change the set of entities the model captures at level N. Thus, the combinatorics of these entities, i.e., the state space, is not changed either.

Type-1 novelty: innovation

> *Innovation: novelty that* **changes the model**.

Innovation is a change to the model: a change that adds a new type or relationship that conforms to the meta-model, or possibly eliminates an existing one.

Innovation changes the combinatorics and the size/structure of the state space, thereby growing/shrinking the possibilities of variation. Examples of changes to the model include:

- moving the value of a variable outside the range constrained in the model: x is modeled as an integer value initially, then $x := 0.5$
- adding a new dimension of the same type: $x, y : X$, then $x, y, z : X$
- duplicating a gene or chromosome

[6] It may also have no effect at all, as, for example, is the case of a neutral mutation.

- growing a parse tree in Genetic Programming
- adding a new species in an ecosystem
- adding a new product or production technology in an economic system

A question for our specific meta-model is whether an innovation systematically corresponds to an *increase* of the number of entity types. Innovation may correspond to the *addition* of a new species, but this event may be associated with the extinction of other species at the same level, thus reducing the total number of entities at this level. Examples of such a process can easily be found in the economy when a new economic good replaces a previous set of goods (for example, the computer replaced the typewriter and the mechanical calculator). Note that the number of entities cannot be reduced continuously at a given level.

The extinction of a species that provides necessary ecological services is a fundamental change to the ecosystem, and can be modeled as a type of innovation at the ecosystem level caused by the *loss* of a species. However, there is no necessity to remove a type from a model when that type has no instances. If the model is not so modified, the extinction is classed instead as variation, with the value of the population size variable decreased to zero.

Type-2 novelty: emergence

Emergence: novelty that **changes the meta-model** .

Emergence is a change to the meta-model: a change that adds a new meta-type or relationship, or possibly eliminates an existing one.

Potential examples of emergent phenomena include:

- the addition of the concept of discrete entities to a meta-model based on continuum concepts;
- the addition of a new mechanisms for variation, such as sexual recombination;
- the addition of the concept of processes to a meta-model based on entities;
- the addition of the concept of energy to a meta-model based on movements.

One particularly important phenomenon is the emergence of a new level of organization. Our meta-model has been designed to highlight this form of emergence, by the inclusion of the concept of discrete levels.

In our meta-model, a change observed at level N cannot be described at this level because it changes the composition rules of the level-N entities. Thus, to be described in our modeling language, this change needs the creation of new entities at level $N+1$ or the elimination of all entities at some level. Note that this can require a reorganization of the levels structure. This is the case, for example, when bacteria and a multicellular eukaryote engage in an endosymbiosis relationship: if the multicellular eukaryote is level-N, the compound is level-$N+1$ and the ecosystem, formerly level-$N+1$, is now level-$N+2$. Or, if the loss of a species at level-N causes the loss of the ecosystem at level-$N+1$, then any structure at level-$N+2$ becomes level-$N+1$.

This generally happens when a population of entities (aggregate system) becomes a new system entity at a higher level. Examples are:

- when configurations in cellular automata exhibit large-scale coherent patterns (see "Illustrative examples")
- when bacterial cells evolve into eukaryotic cells with cytoplasm, nuclei and organelles
- when single-celled organisms form organized multicellular structures such as animals, plants and fungi
- when solitary individuals become colonies or societies
- when a population of cell tissue forms an organ
- when a group of individual economic agents pool their resources and form a company
- when a number of species form an organized ecosystem
- when a number of artists form a new school

We have specifically included each new level as a separate component in our meta-model, ensuring that each transition to a new level is a type-2 novelty. Consequently, each new level will be added to the meta-model only once, and emergence corresponds to the *first* occurrence of the type of entity. Once the first level-$N+1$ entity has emerged as a type-2 novelty, similar entities appearing later will be added to a pre-existing level. Thence, the next level-$N+1$ entity to appear is only a type-1 novelty, an innovation at level $N+1$. Indeed, it "simply" changes the level-$N+1$ structure, but no further change to the meta-model is needed.

Classification versus measures

We have provided a classification of novelties, with respect to some model and meta-model. Our classification complements various measures of novelty in the literature, for example, those of Bedau and Packard (1992), Bedau et al. (1998), Droop and Hickinbotham (2012). Those measures tend to be expressed in terms of 'components' (be they genes, individuals, or whatever); they track abundances of different components as new components enter the system. It is assumed that it is possible to distinguish and compare these new components.

Our classification allows us to determine if the components differ through variation (if they are mutated strings, say), or through innovation (if they are new species to the model, say), or differ through emergence (if they are new levels of organization, say), dependent on the model in use (in the current literature, the relevant model is usually implicit). The cited measures could then potentially be

used to quantify the respective amounts of variation, innovation, or emergence activity, to determine if the activity of the respective type is continuing (see also "Open-ended events and open-ended systems").

Events and classes of events

Any particular change is potentially the result of a set of changes at a range of levels, and may result in changes at a range of levels. We group such changes into an *event*: a set of cascading changes at various levels of the system that ultimately corresponds to a single novelty at a given level. It directly follows from this definition that events can be classified into three classes according to the highest type of novelty they generate at the different levels where they are observable.

The distinction between *changes* (observed at a given level) and *events* (groups of changes at all levels) is a fundamental one. In particular, it captures the question of timescales (see "Timescales") since a single change at one level may be the consequence of many changes at the lower levels. For a given system, the observed events will be relative to the model and meta-model. The composition rules that are used to define levels in our meta-model have no equivalence for the types of novelty: any type of novelty at a given level can be the cause or consequence of any type of novelty at another level. In particular, all kinds of novelty can ultimately be connected to a set of variations at level 0.

Open-ended events and open-ended systems

We are now able to discern three types of events in our model: *variation*, *innovation* and *emergence*. Two of these types, innovation and emergence, we define as *open-ended events*.

> *Open-ended system*: a system with the ability *to continually produce open-ended events*

Variation (type-0) events correspond to the "normal" regime of any dynamical system, and thus we do not consider them open-ended. For example, engineered mechanical systems (for example, a car) undergo events, but none of these events change the structure of the system (i.e., number and kind of entities composing the car).

Innovation (type-1) provides a form of OE of new kinds of things (new types in the model) that nevertheless correspond to existing concepts (the new types conform to the concepts in the meta-model). Emergence (type-2) provides another form of OE, of new concepts (new components in the meta-model). We have designed our specific meta-model to support the idea of "major transition" although this idea is not made explicit here. According to Maynard Smith and Szathmáry (1995), a major transition is the emergence of an entity composed of (sub-)entities that were capable of autonomous replication before the transition but that are no more capable of replication after—while the aggregate entity is now capable of replication. In our meta-model, a major transition is a type-2 open-ended event (emergence) that adds a new level, and therefore the ability to evolve on a higher level. However, the loss of the replication property at the lower level is not included in our meta-model because this property is still present in the low-level entities that are not engaged in this specific major transition. Different changes to this meta-model (adding other concepts than new levels) would probably lead to other classes of emergent novelty.

Interestingly, the distinction between type-1 (innovation) and type-2 (emergence) events captures the distinction between the notion of continual novelty and the notion of continual increase of complexity. The repeated occurrence of type-2 events does not, however, imply that there is an arrow of "progress". A type-2 event implies that new levels must be added to the model, but these can be the cause or consequence of a decrease of "complexity" at other levels through partial or complete loss of structure or function. For example, the establishment of an endosymbiotic relationship between bacteria and multicellular organisms can lead to a heavy loss of complexity of the bacteria (McCutcheon and Moran 2012; Batut et al. 2014).

Limits to open-endedness

Proving that a system will *continually* produce open-ended events is challenging. On a theoretical basis, one can question whether such systems, when instantiated physically, are even possible in a finite universe. A continual production of novelty would require physical systems that are unbounded in space, time, and combinatorial possibilities. Conceptual models and other abstractions are not limited in this way. This point is discussed in "On computational limits" in Appendix.

Should we be looking for systems able to *continually* produce open-ended events, or "simply" for systems able to produce *a sufficient number* of open-ended events? We thus distinguish systems that are *theoretically* open-ended from those that are *effectively* open-ended. The former may be demonstrable in a mathematical universe, but questionable in a finite universe; the latter are questionable in a mathematical universe (at least for non-platonists), but may be demonstrable in a physical universe. If one says that life is open-ended, one can only claim its open-endedness *so far*. Similarly, when seeking to simulate open-endedness most authors are willing to simulate one or a small number of open-ended *events*. This point is further discussed in "Effectively open-ended".

Upward limits

There are severe limits to achieving open-endedness in simulation and also in the real world. The growth of resources required to populate the higher levels will quickly lead to an exhaustion of material or entities. In biological systems, it was this Malthusian limitation that led Darwin to develop the theory of Natural Selection.

A requirement of any such system is a lower level that comprises entities with combinatorial power and behavior, which can generatively combine in multiple ways to form a huge ("effectively unbounded") diversity of entities. We call these entities, and the rules for their combination, the "chemistry" layer. This is level 0.[7] Suppose that, at each level i, m_i entities combine into a single higher level system, and every lower level entity belongs to exactly one higher level system.[8] A level-1 system comprises m_0 level-0 entities. A level-2 system comprises m_1 level-1 entities, each comprising m_0 level-0 entities, hence it contains $m_1 \times m_0$ level-0 entities. A level-k system contains $\Pi_0^{k-1} m_k$ level-0 entities. If we assume for the sake of the argument that all the m_i are identical (and that entities' components do not overlap and are not shared), then such a level-k system contains m^k level-0 entities.

Hence, the number of base entities needed to populate a system grows exponentially with the number of levels. After several levels, a system like this will exhaust the number of level-0 entities available (molecules, computer memory cells). Moreover, for a given m_0 number of atomic entities at level 0 the maximum number of entities at the highest level rapidly decreases as the number of levels grows. As long as Darwinian selection is at work, this decreases the efficiency of selection at this level and can increase the amount of "drift" (Kimura 1984). A direct consequence is that the possibility for the system to acquire a more complex structure (innovation events) also necessarily declines as the number of levels grow. Another consequence is that innovation at each level will be less constrained by selection at that level, and more the consequence of intrinsic dynamics such as drift.

To summarize, the open-endedness of any discrete system with $m_i > 1$ will be limited by the combination of two processes: first, the exponential exhaustion of level-0 entities used in the system's "complexification" limits the possibility of observing type-2 events (emergence); second, the reduction of the selection efficiency with the decrease in number of entities at the higher level limits the possibility of predicting the direction of type-1 events (innovation). Therefore, one can conjecture that a finite discrete system *cannot* be theoretically open-ended (in the sense that it can reach barriers to further novelty that could be transcended only through some "external" intervention to "expand" the finite limits).

Downward limits

Schrödinger hypothesized that living systems were only possible on the macroscopic scale, due to quantum effects at the atomic level (Schrödinger 1944). More generally, he argued that order on large scales was a consequence of disorder on small scales, which he called the "order from disorder" hypothesis. For him, living matter must display homeostasis, which is a maintenance of large-scale (relative to the atomic scale) organization over time. But he also pointed out that living things inherit and bequeath order along their lineage, and that the mechanism for maintaining order through lineages must be small, on the molecular scale. He even postulated that this material was probably an "aperiodic crystal" that conveyed information via covalent bonds. Inheritance on the molecular scale would be a possible explanation of hereditary variation arising, ultimately, from quantum effects. Schrödinger's breathtaking hypothesis prefigured the eventual discovery of the biological function of DNA by Crick and Watson a decade later, and both credited him as an inspiration for their work.

For our purposes, Schrödinger's foray into the molecular basis of life reminds us that innovation is only possible in physical systems that are sufficiently large. If living things were too small, quantum effects would make inheritance impossible: it would be impossible to maintain structure, and innovation is typically only possible in physical systems that are sufficiently large (Anderson 1972).

Effectively open-ended

As explained above, the hierarchical organization of our meta-model imposes strong limits to the open-endedness of any bounded system, particularly when one considers type-2 open-ended events (emergence): the number of levels one can model from a finite number of level-0 entities is severely limited by a potentially exponential increase of level-0 entities dynamically linked in level-N entities. Moreover, if one simply considers *variation* in a state space with a finite number of dimensions, each of which can take a finite number of values (for example, a bitstring which forms a boolean hypercube), then the state space itself is finite and bounded, and so the possible variations are bounded. Strictly speaking, the system is not open-ended. However, considering a system S at a time t of its

[7] With our meta-model model, it is not "turtles all the way down"!

[8] This does not imply or require that the set of m_i entities constituting any specific level-$i+1$ individual is fixed or static for the lifetime of that individual. It is a specific characteristic of biological individuals that they continuously turn over their constituent lower level components, while retaining their systemic coherence and individuality.

evolution, we propose to say that S is *effectively* open-ended if the size of the population at the highest level of the hierarchy is large enough such that open-ended events are still possible.

Physical systems (for example, the biosphere, techno-social systems) are effectively open-ended if one considers type-1 open-ended events (innovation). Indeed, these systems have a sufficiently large state space and the states realizable or explorable within a reasonable time (say, within the age of universe) are an insignificant fraction of the entire set of "interesting" states. When considering type-2 open-ended events (emergence), whether these systems are effectively open-ended is for us quite literally an open question. Indeed, either the biosphere or the techno-social systems are now limited by the finiteness of earth's resources and the possibility of the emergence of new levels of organization is not clear.[9]

Now, consider computational simulations.[10] The number of entities in a simulation is severely limited by computational power, including both CPU time and memory. In the universe, there is a vast number of molecules; in a computer simulation, there is a large, but not vast, number of memory cells (its "effective unboundedness" is less effective). If we wished to simulate interesting models with many levels, we could not do so from the bottom up: there would simply be too many level-0 elements to simulate efficiently, or even to fit into memory. Additionally, regularities in the dynamics of a level-$N+1$ entity comprising a system of level-N entities tend to be on a longer dynamical timescale (slower processes) than in the level-N entities themselves and the events we are seeking are likely to be rare events. This implies the need for an exponential amount of simulated time in a multilevel system. Thus, the upward limits will be quickly reached as the number of levels grows (that is, as emergence events accumulate). One can still simulate *effectively* open-ended systems if the total number of levels in the system is low enough for the number of entities to be large. However, though effectively open-ended, the system will eventually exhaust its possibilities.

[9] One can still argue that, on a larger horizon (namely on the level of the entire universe), although the universe is bounded, the potential for novelty is unlimited, be it generated by variation, innovation or emergence. Following this idea, one could consider that the universe is *effectively open-ended*.

[10] As opposed to statistical simulations, such as "draw an infinite number of reals from (0, 1), with an exponential distribution", where innovation in the form of seeing new numbers is certain. Our argument here applies to any physical simulation, not just to computational ones.

Simulation of open-endedness

Consider artificial life as an example of computational simulation. One of the central goals of ALife is to use simulation to study and understand open-ended systems, including open-ended Darwinian evolution (Bedau et al. 2000). In particular, we wish to study how living systems have undergone successive major transitions from the first replicative molecule to the techno-social level (Maynard Smith and Szathmáry 1995). Systems of this kind have a hierarchy of levels, with lower level entities forming higher level systems. Thus, this goal directly hits the computational limits we conjectured above and may be an inaccessible dream, at least with current computational technologies. Hence, we need to optimize the approach, taking stock of the exhaustion of resources. What that means is that we need to hard code some of the emergent properties and laws at certain levels, rather than letting them emerge from lower levels. Some hard-coding will cover the entire set of lower levels not considered (the "generative" layer), and some will cover the examined (emergent) levels to "cheat" (optimize, hard-code) certain behaviors through what we call *shortcuts*.

Generative layer

The question of level-0 entities in natural systems is an open question: is level 0 populated by molecules, atoms or even lower level particles? Now, in a simulation, one cannot start from too low a level, simply because it would naturally increase the number of levels that separate level 0 from the level of interest for the question under study. Thus, our multilevel model will generally possess a lowest combinatorial level (level 0), which we refer to as "chemistry" or "artificial chemistry" (Dittrich et al. 2001; Banzhaf and Yamamoto 2015). This level should be able to subsume all the diversity created by putative lower levels (atoms, particles, and below) and, conversely, provide the necessary diversity of elements to the higher levels. This basal level provides the "physics" (rules and behaviors) and "chemistry" (combinatorics) of the simulation, thus implementing level 0. For example, in biological evolution, the existence of elements that can be combined through bonding dynamics into molecules of different types—in itself a substantial compression of what goes on in atom/quantum physics and quantum chemistry—provides different chemical characteristics as a prerequisite for the variety we can observe at higher levels of the hierarchy.

Examples of generative layers are numerous in the ALife literature. It can be composed of bit strings as in the case of genetic algorithms [for example, Holland (1975)], artificial molecules in artificial chemistry [for example,

Hutton (2002)] or code in genetic programming [for example, Koza (1992)], to give a few examples.

Shortcuts

If one wants to study a layered system like the ones we envision in our meta-model, one needs to simulate multiple levels that successively emerge from the generative "soup" of level 0. As conjectured above, such a simulation is impossible to implement in practice for large values of N. We may thus need to introduce new elements in our simulation that will simplify or accelerate the dynamics of the higher levels. We call these new elements *shortcuts* since they directly implement some properties of the higher level that, in an ideal simulation, would emerge from the generative layer.

Shortcuts are hard-coded design optimizations manually introduced into the simulation. One can consider the shortcuts as "cheats", which limit the generality of the model. It would certainly be preferable to be able to simulate open-ended systems (or at least effectively open-ended systems) without making use of any such stopgap measures. This is clearly feasible, but the number of emergent levels in the simulation will definitively be limited and this would probably preclude the simulation of deep multiscale models,[11] such as those ranging from molecules to ecosystems. If one wants to simulate multilevel systems with a large enough number of levels, shortcuts will be necessary and, to the best of our knowledge, all ALife simulators designed and implemented so far have used some kind of shortcuts. It is better to be aware of this fact and to use it explicitly, rather than to introduce shortcuts implicitly in a simulation without even realizing it.

The problem of how to try to simulate OE can be reformulated as: how to shortcut in a principled way? or how can we design shortcuts that provide optimizations, without precluding the possibility of particular kinds of OE of interest. Indeed, shortcuts provide optimizations by explicitly constraining structures and behaviors at their level, rather than requiring these constraints to emerge from the system's behavior. Hence, specific shortcuts will enable or constrain certain classes of OE.

When designing the simulation of an open-ended system, one crucial design step is then the identification and implementation of relevant shortcuts. These will be research dependent, i.e., they will depend on the specific question one wants to answer with the simulation. Examples of such potential shortcuts that might be inserted in certain simulations include *individuality*, *replication*, and *fitness*.

Individuality

Individuality is the "mother of all shortcuts". Entities of a level $N > 0$ are hard coded rather than emergent. This can be implemented in two slightly different ways: either by directly simulating the dynamics of these entities, which could be called an "identity shortcut" or by setting boundaries that group together entities of level $N-1$ in level-N entities, which would rather be a "boundary shortcut". In Fig. 2, an identity shortcut hard codes the circles of subfigure C while a boundary shortcut hard codes the dashed circles of subfigure B. Obviously, this shortcut makes it impossible to observe emergence at level N but it still enables emergence at level $N+1$. However, when limited to the boundary part, an individuality shortcut enables innovation at level N.

Replication

Replication is a central process in biological evolution. In computational simulations, a replication shortcut is the explicit implementation of operators that replicate the entities at level N from the outside of the entities. This shortcut generally comes with secondary ones, for example, to synchronize the replication of all the entities of the same level or to include variation operators that slightly modify the offspring.

Fitness

Fitness in evolutionary computation is a classical shortcut. It replaces the differential reproductive success emerging from the difference between level-N entities (in terms of composition of level-$N-1$ entities), and from their interactions in the level-$N+1$ population, by an explicit computation of reproductive success according to some target task such as computing a given function, seeking "food", gaining "energy", and so on.[12]

Many authors have argued that explicit fitness precludes OE (Channon and Damper 2000; Baptista and Costa 2013; Ray 1992) and some have proposed to replace it by *implicit* fitness or by subtler approaches such as selection for novelty (Lehman and Stanley 2011; Soros and Stanley 2014). This is sometimes merely a change in terminology, renaming the target task with something more biologically

[11] A two scale model is technically "multiscale", but can perhaps be simulated in some cases.

[12] Fitness in biological systems is *defined* as differential reproductive success. But in nature fitness is *assessed* retrospectively, through natural selection.

sounding, such as replacing "increasing a score" by "accumulating energy".

However, OE should be possible with an explicit fitness at a level N (which would shortcut differential reproductive success driven by interactions at level $N-1$), providing that level N entities interact in such a way that level $N+1$ can still have an influence on their reproductive success. In effect, the variations at level N (the phenotype) may induce variation at level $N+1$ (the ecosystem). The fitness values of level N entities, though explicit, will then change due to the downward causation of level $N+1$ on level N (that is, the same phenotype could have different fitness depending on its neighborhood). This could happen, for example, when the model includes an explicit fitness for resource uptake but that this resource is shared by all individuals [which is, for example, the case in Tierra (Ray 1992)].

A general architecture for open-ended simulations

In Fig. 5, we identify a general architecture for layer-based emergent simulations of OE. This architecture separates out the base layer, implementing the physics and chemistry, from the shortcuts at higher levels. The base layer of "generative rules" provides the implicit coding of all levels below the lowest simulated level. These rules generate and enable the lowest level of simulated entities, and the relevant lowest level of an internal environment.

In addition, each simulated layer may have some hard-coded shortcuts. The collection of these is the set of "structuring rules". These rules provide optimizations by explicitly constraining structures and behaviors at the level, rather than requiring these constraints to emerge from the system's behavior. In particular, these rules might provide shortcuts for determining or recognizing the identity of entities, for communicating between entities at a level, for providing parts of the internal environment, and for explicitly implementing "downward causation"-style constraints imposed by higher levels of the system.

Note that for population systems in this architecture to be at a higher level than their constituent entities, they cannot be mere aggregate systems (which do not have a level in our meta-model) but must be genuine system entities. If we were to consider a population-as-aggregate system to be at a higher level than its constituent entities, then we could *never* get interesting emergence (in this meta-model).[13] This is because we would generate the higher level (and get emergence) merely by virtue of having the population; there would be no opportunity to generate the (emergent) new level later, when the higher level system entity itself arose. Making a population into a system entity immediately is a form of shortcutting an emergent transition.

Changing the model

The previous architecture provides a methodology to *simulate* open-ended systems by means of software components that are not open-ended. However, if one wants to use OE to create software technologies to implement creative machines, a.k.a. computational "living technologies", one may want to implement OE directly at the software level. But how can changes working outside the model be implemented in software? In conventional programming languages, the code is fixed, so the design model is fixed. All that happens is that the pre-existing code executes. Generally, no new code is produced. In some languages, however, it is possible for code to modify itself as it runs.[14] In standard software development, self-modifying code is generally to be avoided, as it makes it difficult if not impossible to know beforehand what the code will do; therefore, very difficult to design (or debug) code to demonstrate specified behaviors. Developers usually write code to do something specific and well defined (such as calculate payroll, control power stations, or launch small irate birds at acquisitive pigs). In low-level assembly languages, self-modifying code is relatively easy to achieve. "Automata chemistries" (Tierra, Avida, Stringmol) are examples from ALife that exploit this concept, constructing entities that are strings of assembly language instructions, hence new code (see further discussion in "Coreworlds/automata chemistries"). Some higher level languages have also been designed explicitly to facilitate program self-modification (Spector and Robinson 2002).

In most high-level (structured) languages such as C++, Java, Python, self-modification requires special effort and is not possible in all cases (other than using the language to implement an interpreter for a new language in which it is possible). A "reflective" language can examine its own code; some languages (particularly interpreted rather than compiled languages) can also produce new code on the fly. Such a language is needed to implement software that can change its own model as it runs. The final model of the system, implicitly produced by executing the software, will need to be inferred.

[13] This is another possible difference between our computer simulation-motivated meta-model and biology. This argument in a biological context would imply that innovation in living systems is impossible without group selection. This would be a highly contentious claim.

[14] This is different from external modification, or "patching", running code.

Capturing the novelties

The shortcuts and corresponding simulation code above explicitly implement the (computational) model of the simulation. To capture novelties that change the model (innovations) and meta-model (transitions), there needs to be simulator code to make these changes. There are several possibilities:

1. The emergent novelty is recognized *outside the simulation*. The simulation data are analyzed and the presence of relevant emergent "model-breaking" phenomenon is inferred, but not explicitly realized as a new entity or level in the simulation (for example, the emergence of a flock in a boids simulation (Reynolds 1987), recognized through visualization by the observer or by statistical analysis, but not explicitly present in the simulation code).
2. The type of emergent novelty is *anticipated and recognized*, with pre-coded recognizers present in the code, available to report it once it appears (for example, a boids simulation with a pre-coded flock detector that recognizes if and when it emerges).
3. The type of emergent novelty is *anticipated and captured*, with pre-coded structural rules present in the code, available to recognize and make it a component of the simulation once it appears.
4. The emergent novelty is somehow *emergently recognized* by the simulator, and new code is generated by the simulator to capture the recognition [for example, self-modifying code (Stepney and Hoverd 2011)].
5. The emergent novelty is somehow *emergently captured* by the simulator: once recognized, new shortcut code is generated by the simulator to capture the specific emergence [for example, self-modifying code (Stepney and Hoverd 2011)]. The difference to the previous situation is that the new shortcut here is not only detected as an event but shortcut code is also created that accelerates the simulation by explicitly coding the new, emergent, level.

Design principles

We can use the CoSMoS (Complex Systems Modeling and Simulation) approach (Andrews et al. 2010, 2012; Stepney 2012; Stepney and Andrews 2015; Stepney et al. 2016) to help implement a principled computer simulation that conforms to the architecture described above. CoSMoS is a particular approach to simulation of complex systems and their emergent properties, but the components are necessary (if not always explicitly identified) in any principled simulation work. The components include:

- Research context: a description of the research question(s) to be addressed by the simulation, for example, "to understand the system's exploration of its designed search space", or similar.
- Domain model: a scientific model of the biological, socioeconomic or other system of interest, including the emergent levels and OE classes of interest (for example, speciation, or a major transition, in an evolutionary model), such as shown in Figs. 2 or 4.
- Platform model: an engineering model of the software simulation system. It does not include a model of desired emergent properties, explicitly ensuring that the answer is not coded into the software. It includes, however, additional features needed for efficient simulation, such as the generative and optimization shortcut components in our proposed architecture (Fig. 5), and interfaces.
- Simulation platform: the software implementation, built from the Platform Model specification, suitable for running Simulation Experiments.
- Results model: a scientific model analyzing the output of the simulation experiments in terms of Domain Model concepts.
- Argumentation: that the simulation is "fit for purpose" relative to the Research Context; specifically here that the shortcuts neither inhibit nor hard code the desired OE in the simulation platform.
- Meta-model: a model encompassing the concepts used in the Domain, Platform, and Results Models, to ensure consistency between these concepts across the process.

Within the CoSMoS approach, the framework described in this paper gives a structured way of thinking about the design of a system to display open-ended novelty, including the model, any simulation of it, and the effect of particular design decisions.

For example, we could design a simulation to investigate our open-ended novelty framework within the CoSMoS approach by producing the following components (summarized in Fig. 6):

1. A meta-model: given that our framework defines novelty with respect to a model and a meta-model, it is necessary to define a particular meta-model. The meta-model defined in this paper (Fig. 3) is one possibility; others might be used to capture other forms of OE. Whatever meta-model is chosen, it must be sufficient to support the desired concepts of OE.
2. A domain model: an instantiation of the meta-model in domain terms, including desired emergent properties (where known), and excluding shortcuts.
3. A platform model: another instantiation of the meta-model related to the domain model in the following way:

- the domain desired emergent properties are not present in the platform model, to ensure that they do not become explicitly coded into the simulator
- once the necessary shortcuts have been determined, and have been argued not to preclude the desired emergence, the relevant domain components are replaced with their shortcut alternatives in the platform model
- the remaining domain model components are transferred across to the platform model
- instrumentation and interfaces are added, to allow the simulator to be used to run open-ended novelty experiments; this might include emergent recognizers and model changers (see "Capturing the novelties"), where appropriate.

4. A simulator built to the specification that is the platform model.
5. A results model: a further instantiation of the meta-model, suitable for capturing and analyzing observed emergent properties of the simulator in domain model terms.

The CoSMoS approach is a principled way to examine complex systems. There are other possible approaches that can be used to avoid confusion and to make sure that the novelties discovered in a simulation were not already built into the system from the outset. One should use some principled, coherent methodology, so that one can make meaningful claims about types of novelty.

Illustrative examples

In this section, we explain via examples from the literature how the model/meta-model approach captures the presence and absence of OE in both computational systems (Game of Life, "Game of life") and real-world systems (experimental biological evolution, "Experimental evolution"), and in particular how it is all *relative to a model/meta-model*.

Then we give examples of genetic algorithms ("Genetic algorithms"), genetic programming ("Genetic programming"), and automata chemistries ("Coreworlds/automata chemistries") to demonstrate some of the "shortcuts" described earlier, how they constrain the amount of novelty, and what consequences these shortcuts have for OE.

Game of life

Conway's Game of Life (GoL) (Berlekamp et al. 1982; Gardner 1970) is a two-dimensional cellular automaton (CA), and a typical example quoted when talking of novelty and emergence. Here we show how GoL exhibits various kinds of novelty relative to a model instantiated from our meta-model. Other authors discuss emergence in GoL in different terms, for example Gotts (2009).

The base-level model has the following components: the cells, *atomic entities* at level 0 that can be in state 'alive' or 'dead'; the entire GoL arena, an *aggregate system* comprising a collection of cells laid out in a grid pattern; the *environment*, considered as providing the space and time for this grid; and *interactions* between cells, which communicate their state to their eight nearest neighbors, and behave by synchronously updating their own state based on the GoL rules: each timestep: if a cell is dead and has exactly three live neighbors, it becomes alive; if it is alive and has fewer than two or more than three live neighbors it dies.

As the CA executes, cells change state, exhibiting *variation*, that is, type-0 novelty (a change within the existing model only modifying the value of the existing state).

Rapidly, the observer notices the appearance of certain patterns: some localized groups of cells switch through cyclic patterns (on a longer timescale). These localized groups are modeled as level-1 entities, comprising the relevant level-0 cells and the local grid environment over an extended number of update cycles. Thus, the meta-model is augmented with a new level: level 1; the model is augmented with a specific system entity type: an oscillator (including period-1 oscillators called "still lives"). This modeling of oscillators is then an example of *emergence*, that is, type-2 novelty (a change to the meta-model adding a new level).

The observer also notices further patterns that seem to move across the grid, called "gliders" (or, more generally, "spaceships"). These are also modeled as level-1 entities, comprising the now-changing sets of cells and local grid environments that encapsulate these patterns. This modeling of gliders is then an example of *innovation*, that is, type-1 novelty (a change to the model adding a new entity type).

Then, the observer notices that these moving gliders *interact* by colliding: such interactions can produce new oscillators, gliders, and other patterns. This modeling of gliders is also an example of *innovation*, that is, type-1 novelty (a change to the model adding a new kind of entity, hence a new interaction type).

Eventually, the observer notices that these moving gliders and oscillators can be positioned to interact in such a way that they form a level-2 entity: a Turing Machine (Rendell 2002). This model of Turing Machine in the GoL is then another example of *emergence*, that is, type-2 novelty (a change to the meta-model adding yet another new level).

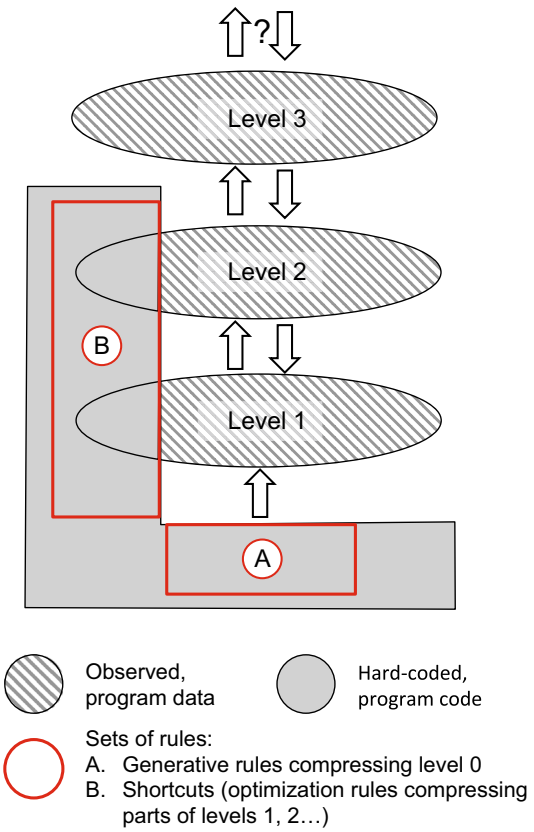

Fig. 5 Simulation with shortcuts, generic figure. Level 0 is abstracted by the set of generative rules encoded in "A". Levels 1 and 2 are predefined by the simulation (parts of them are hard coded) but they include emergent parts that enable variation, differences or innovation. Level 3 is fully emergent, not at all included in "B" (but may be anticipated by the modelers when designing "A" and "B" since they want to observe 3). If level 3 is a system made of entities of lower levels, then its emergence is a transition. If the level 3 entities are replicators, then emergence of level 3 is a major transition

In this example, the scientific model and meta-model are changed based on observations, hence innovation and emergence are observed. The possibility of innovation and emergence spring from the GoL engineering model that is itself not changed during the process (see Fig. 1).

Experimental evolution

Experimental evolution is a biological discipline where organisms evolve in controlled experiments, to explore natural evolution in real time. By defining experimental protocols, and preparing isolated strains of organisms for observation, shortcuts are introduced into the process, removing it from the natural evolutionary environment.

Since many predictions of evolutionary biology assume very large effective populations or many generations, most experimental evolution systems use viruses or bacteria, though other microorganisms (such as yeast and other fungi, nematodes, and more) and even some larger organisms (including fruit flies, mice, plants, and more) have also been used. The typical experiment involves several generations of the organism(s) under study, while selecting for some trait. Given a response to selection, which is nearly always present, the biologist can untangle the causes of the response using genetics, transcriptomics, metabolomics, or other techniques.

Experimental evolution is often "open-ended" only in that the experimenter does not specify or enforce pre-determined mechanisms for adaptation. However, this is not the kind of open-endedness we defined in this paper. Yet, innovation—in our use of the term—commonly emerges, such as when viruses adapt to a new host, new temperature, or other environmental change in an evolving viral/bacterial system. Some experimental evolution systems, such as Richard Lenski's famous "long term evolution experiment" have shown a clear open-ended dynamics (in our use of the term). Lenski has transferred over 62,000 generations of *E. coli* as of the time of this writing (personal communications), without (intentionally) selecting for any particular phenotype (Barrick et al. 2009). Different lineages of organisms with clearly distinct phenotypes and reproductive fitness have emerged during the course of this experiment. More recently, after over 27 years, a genuine speciation event may have arisen, where the bacteria have developed the ability to metabolize an entirely new energy source, which was always present in the media but never used: citrate (Barrick and Lenski 2013; Blount et al. 2008). Other open-ended events have been observed in this experiment such as the observation of a stable polymorphism (Plucain et al. 2014) that may correspond to the emergence of a new ecological level. One of the main difficulties in experimental evolution is to understand how the "shortcuts" (e.g., serial transfer, culture medium, artificially stable environments...) are likely to trigger the observed OE events.

Genetic algorithms

The genetic algorithm (GA) (Holland 1975; Goldberg 1989) is another example of the benefits of shortcuts. GAs have exhibited an astounding ability to produce creative and surprising solutions to technical optimization and design problems (Renner and Ekárt 2003).

GAs work by manipulating a set of parameters that code for the solution of an application problem, such as the minimization of a function. Each of these parameters might be considered a level-0 entity, with the set of parameters providing a solution being an entity at level 1. In GAs, this is called a *genome*.

At a still higher level in the GA (level 2) sits the so-called *fitness function*, a shortcut for helping to judge the

quality of a solution. In biology, fitness is an implicit concept assigned retrospectively to individuals based on their reproductive success. In GAs, fitness is usually explicitly defined, while individuals are implicitly assumed to be produced by the genome. We described evolutionary computation above in terms of function minimization, where the difference between the function and what is required for the minimum (the *error*) is equivalent to (inverse) fitness and can be used to judge solution quality.

The list of fitness values can now be used to provide a selection signal from this level-2 system to level-1 entities by removing low-quality solutions and thus providing space for new individuals in the population generated by variation processes. The actual variation is produced by random change on individual parameters (*mutation*) or by choosing sets of parameters from different individuals (*recombination*).[15]

Often a set of discrete parameters can encode a solution for the problem and the number of dimensions does not change. Thus, the space of all possible combinations of parameters (and therefore solutions) is bounded. Yet, still interesting and novel solutions might emerge (despite using only type-0 novelties) as the combinatorial spaces under consideration quickly exceed the number of elementary particles in the universe. The limiting factor in GAs seems to be the fixed fitness function, which leads the search sooner or later to stagnating fitness plateaus.

The situation with respect to shortcuts is depicted in Fig. 7. A genetic algorithm as well as other similar evolutionary systems are highly constrained. Individuals (level 2) are organized in a population (level 3) with an entirely hard-coded structure that imposes selection on individuals. Individuals are made of data but some of their properties are hard coded (individuality, replication). Individuals are made of genes that are chemical entities (level 1) producing the variation. Here, however, variation is hard coded and cannot change. The generative rules (denoted by "A" in Fig. 7) are also highly simplified: they are a fixed-size bitstring.[16]

Genetic programming

Genetic programming (GP) (Koza 1992; Banzhaf et al. 1998) is a computational method using similar principles as GAs. The individuals under variation and selection are computer programs or other active entities. Here, type-1 novelties make their regular appearance, as a result of the active nature of the required solution. Still, an explicit fitness function (often unchanging) governs the procedure and leads to improved solutions. Individuals in GP have to accommodate regular dimensional changes and should be better considered *behaviors* of their genomes (Foster 2001).

Level-0 entities in this case are elementary behaviors that might be useful in the context of the problem. An example might be a GP system learning to play a game like chess. The combination of elementary behaviors makes up an individual able to play a game of chess. This would include the judgement of particular positional situations which would lead to different responses. These game-playing strategies would be level-1 entities subjected to type-0 and type-1 changes, again controlled by random mutations and recombinations.

Level 2 would be a more or less successfully played game. Here we can see that higher levels require the integration both in time and space of what is coded in the genome of an individual. Again, the constraints to the GP model are severe and tend to greatly limit the emergence of novelty.

In some forms of genetic programming, however, the mechanisms of variation are not fixed. For example, in "meta-genetic programming" a population of variation operators co-evolves with the main population (Edmonds 1998; Kantschik et al. 1999), while in "autoconstructive evolution" systems the individuals in the main population are responsible both for problem solving and for producing offspring with variation (Spector and Robinson 2002;

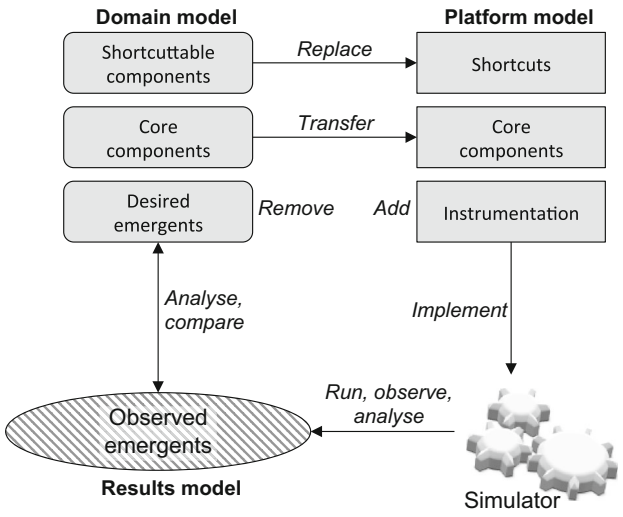

Fig. 6 Open-ended novelty simulation design within the CoSMoS approach. The Domain, Platform and Results models are instantiated from a common meta-model, and related to each other as shown (see text for further explanation)

[15] The variation is technically *pseudorandom*, being generated by a deterministic algorithm. This process is itself a lower level system in the model. Details of this level can significantly affect GA behavior.

[16] Note that open-ended GAs could be implemented providing the bitstring length and the number of genes it encodes can evolve. In this case *innovation* events could be possible as observed, e.g., in Knibbe et al. (2007).

Spector 2010). In these systems the mechanisms of variation are themselves subject to variation and selection, and hence can evolve. The potential that these systems have for producing open-ended novelty is a subject for future study.

Coreworlds/automata chemistries

So-called "coreworlds" or "automata chemistries" denote a loosely defined set of related, but distinct, evolutionary computational systems. Examples include Coreworld (Rasmussen et al. 1990), Tierra (Ray 1992), Avida (Adami and Brown 1994) and Stringmol (Hickinbotham et al. 2010, 2016). These are conceptually, or superficially, similar to GP systems, in that the Darwinian individuals are computational processes (executing programs) hosted in some shared, or at least interconnected, memory system.[17] However they should be contrasted with GP systems in a number of important respects.

First, replication is not implemented by shortcut: the processes are responsible for their own replication. Indeed, non-replicating or sterile processes can be instantiated by design, or can arise through spontaneous perturbation, and can significantly influence evolutionary dynamics; a phenomenon which is not possible in most GP (or GA) systems where all individuals are directly replicated by shortcut, regardless of their intrinsic functionality. As one specific consequence of this *self*-replication, fitness is, to some significant extent, intrinsic rather than extrinsic.

Second, the individual processes execute (pseudo-)concurrently, and their dynamic interaction typically constitutes another significant factor determining relative fitness.

Third, the programming formalism is usually based on a machine code representation (translated to an assembler-like representation in the user interface); as opposed to the high-level formalisms (e.g., `lisp`) typically adopted in GP. This formulation makes self-modification/metaprogramming (the ability of programs to treat their own code, and that of other programs, as data to be operated on) very natural and easily accessible. In almost all empirical work with coreworlds, this metaprogramming forms the basis for implementing replication through a mechanism of direct self-inspection. However, this is not in any sense intrinsic to the underlying platforms, and some examples exist in the literature of implementing fully von-Neumann style self-reproduction with mutable GP-mapping instead (McMullin 2012; Hasegawa 2015; Baugh 2015).

In terms of the framework presented here, all coreworlds have a base-level model (level-0) which includes one or more discrete random access memory system(s) and a dynamic (and indefinite) number of execution threads (CPUs). At this level, they generally support type-0 novelty through stochastic perturbations of memory contents. An individual process (level-1) is constituted by one or more CPUs coupled with some configuration (content) of one or more segments of memory. All coreworlds directly allow for type-1 novelty at level-1 (the spontaneous and open-ended introduction of new types of level-1 entities). This is typically triggered through the type-0 stochastic perturbation of memory contents at level-0; but can also be generated through (re-)writing the content of memory locations (i.e., by programmatic action). Either type of mechanism can lead to expansion (or contraction) of the memory segment(s) in use by any given process, that is, even if level-1 individuals are modeled as dynamic systems on a state space, the dimensionality of this state space can itself change.

Level-1 individuality in coreworlds (individually distinguished and identified executing processes) is normally shortcut to the extent that there is external "operating system" support to group together one or more execution threads with one or more memory segments, which the system monitors and logs as constituting individuals. However, the coherence of this shortcut may be blurred or undermined if the memory segments are drawn from a shared address space, because in that case individual processes may still potentially read, write, or execute memory segments that are *notionally* associated with a different process (or with none). This can have significant phenomenological consequences in particular coreworld systems.

Typical experiments in these systems are externally seeded with one or more such (level-1) individuals, which have been designed to have the property of self-replication. Further, the self-replication mechanism is such that it supports the possibility of heritable mutation (type-1 novelty within the class of self-replicating level-1 individuals). Thus, the meta-model already allows for the development of populations (lineages) of level-1 individuals having essentially identical programmatic behavior (by virtue of identical initial memory content). These lineages are then level-2 entities that can be classified into distinct strains, which exhibit typical ecological and evolutionary phenomena. However, because these strains were already part of the meta-model, their appearance does not generally constitute type-2 novelty (*emergence*).

That said, an exception might be argued for the `Amoeba` coreworld system (Pargellis 2001). In that case, it is not seeded with self-replication functionality at level-1, but that spontaneously emerges. The system was conceived and designed with that specific possibility in view, as have been several autoconstructive evolution systems (Spector

[17] The "core" in "coreworld" derives from the earlier `CoreWar` programming game (Dewdney 1987), and invokes the typical linear, random access, memory configuration originally associated with early magnetic core hardware.

and Robinson 2002; Spector 2010). Nonetheless, it can be said that the meta-model initially lacks any coherent level-2 entities; but once the level-1 self-replication functionality emerges, then such population-lineage entities must be added to the meta-model (i.e., level-2 must be introduced) and, therefore, this is properly an example of type-2 novelty.

Returning to the case of coreworlds where self-replicating level-1 individuals are externally seeded, there are still some subtleties in classifying the types of novelty arising. In the case of the "original" `Coreworld` (Rasmussen et al. 1990), although seeded with coherent self-replicating level-1 individuals, the self-replicating functionality (and thus conventional Darwinian evolutionary dynamics at the level-2 of interacting lineages) is typically lost very quickly, due to unrestricted overwriting of memory segments notionally associated with different individuals.[18] This could be modeled as a type-2 novelty where the meta-model is changed by *losing* a level. By contrast, `Tierra` introduced a critical design feature of a "write protection" shortcut. This allows individuals to protect against external disruption of their own "allocated" memory segments, while still allowing them to pervasively read the contents of all memory (allocated or not). This ensures reliable persistence of the self-replication functionality and thus the level-2 of distinct strains. Unfortunately, in itself, this also allows a single self-replicating seed process to generate an exponentially growing lineage that quickly exhausts (allocates) the entire, finite, memory system. This would prevent any further replication (since no more memory segments can be allocated), and bring evolution (open ended or otherwise) to a halt. Accordingly, it was necessary to introduce a further, additional, shortcut, the so-called "reaper", which stochastically "kills" random processes, and deallocates their memory segments (and destroys their execution threads). This does allow `Tierra` to exhibit sustained type-1 *innovation* at the level-2 of strains, including phenomena of selection and diverse forms of parasitism.

In practice, however, evolutionary novelty even in `Tierra` quickly "plateaus". While there is continuing generation of type-1 novelty at the level-1 of individuals, and this does continue to generate new, distinct, strains at level-2, these seem to exhaust the scope for any further distinctive ecological or selective behaviors at level-2, that is, there is no further even type-1 novelty at level-2, and certainly no generation of identifiable, coherent, entities at any higher level (i.e., no instance of type-2 novelty of *emergence*).

[18] Strictly, `Coreworld` does not incorporate a memory "allocation" shortcut per se at all.

The `Avida` system sought explicitly to open up further evolutionary potential in a coreworld system (as compared to `Tierra`) by adding the possibility for additional, externally specified, "behavioral challenges". These take the form of "tasks" that level-1 individuals may complete, and by doing so may enhance the Darwinian fitness of their corresponding lineages (strains). While this does lead to a diverse range of interesting phenomena, for the purposes of the current discussion, it does not seem to add any decisive new feature. In particular, the "plateauing" of novelty phenomenon is still observed. It is delayed somewhat while solutions are sought for the externally defined tasks. Once those tasks are satisfied, evolution again stagnates as, within the system itself, there is no mechanism for new tasks to arise and, as with `Tierra`, there is no case of type-2 novelty (emergence of level-3 entities).

For the moment, it is an open (and active) research question whether the coreworld approach can be further developed in a way that would significantly extend its capability to exhibit open-ended evolution. As a minimum, any such development would seem to require mechanisms for generation of new selective "niches" (i.e., where selection, at the level-2 of strains, is not operating in just a single fitness landscape) and for the potential "individuation" of new entities at level-3.

Conclusions

Summary

In brief summary, our argument is:

- Models and meta-models are used for building both scientific and engineering models of systems (they may be implicit)
- We define types of novelty and open-endedness *with respect to the system's current model and meta-model*:
  - Novelty in an observed system is classified as:
    0. Variation: novelty within the model
    1. Innovation: novelty that changes the model
    2. Emergence: novelty that changes the meta-model
  - *Open-ended event*: an event that results in innovation or emergence
  - *Open-ended system*: a system with the ability *to continually produce open-ended events*
- We provide one particular meta-model, incorporating the concept of *levels*, which allows it to capture "major transitions" as emergent events

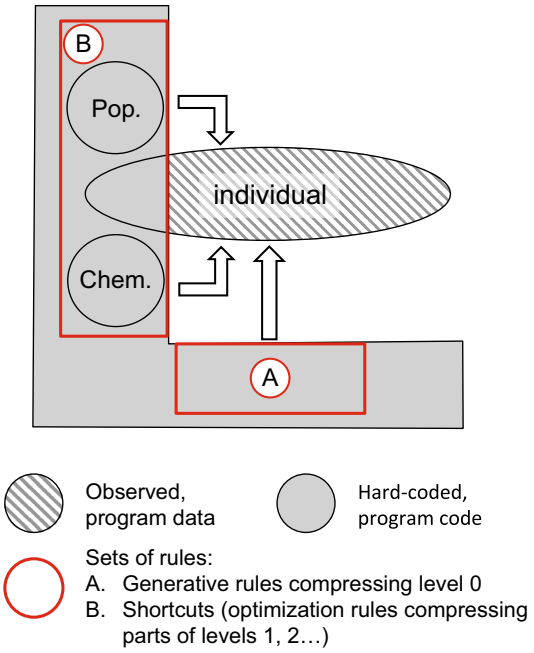

Fig. 7 Simulation with shortcuts, GA example. In a GA, the level of interest is the individual. These individuals are grouped into a population and they are made of elements (genome, phenotype) that give them their dynamic. Now the population level is a mere aggregate of individuals (see Sect. "A general architecture for open-ended simulations"), hence not an explicit level in the hierarchy. Similarly, the elements that compose the individual and their relationships (e.g., the genotype-to-phenotype mapping) are explicitly encoded in the software. So the classical GA structure is a level 2 composed of individuals that evolve through variation events imposed by a fully shortcut lower level and a selection pressure imposed by a fully shortcutted higher level. Note that for sake of clarity we distinguish the generative rules (**a**) from the lower level chemistry rules in box (**b**) but that the latter could have been merged with the former

- We introduce simulation "shortcuts": hard-coded behaviors at different levels needed to implement effective computer simulations
- We describe some illustrative existing systems in terms of their types of novelties and shortcuts

Discussion

The constant emergence of novelty from complex systems is a fundamental phenomenon of nature, societies, and computer simulations for which one might desire a unified theory. However, as we have shown, it is difficult to precisely characterize a notion of open-endedness that works in all cases, so that a general theory of open-endedness is hard to come by. The barriers are both conceptual and computational, as we have highlighted in the appendices. However, we have argued that it is possible, and even useful, to attempt such a general theoretical framework, without committing *a priori* to any particular application domain.

One function of a general theory would be to develop domain-independent models of open-endedness, so that predictions and experiments in one domain might support cross-domain conclusions. We have presented a meta-model for open-endedness that we have argued can encompass similarities between models of open-ended evolution in biological systems, the emergence of socio-economic systems, and computer simulations. We have applied this meta-model specifically to the unbounded emergence of novelty in computer simulations, with sufficient software engineering detail to show that our theoretical aspirations do not sacrifice practical applications.

Our emphasis on computer simulations is a pragmatic choice. Suppose we had chosen biological evolution to make our argument. Life on earth has demonstrated continuing emergence of novelty, but it has taken billions of years and a vast number of molecules and organisms, not to mention the abiotic and energetic richness of an entire planet, to do so. While a general theory should be able to encompass this grand sweep of biological complexity, it would be very difficult indeed to test. The best one could hope for would be to develop models of open-endedness that predict, and hopefully predict the parameters for, specific biological phenomena.

In any case, biological models alone are insufficient; models of biological innovation should be paired with laboratory or field experiments. This has been done in some specific domains, such as the emergence of cooperation, resource competition, virulence, and collective behavior. Much of this work has been done in microbial systems such as bacteria and yeast, since the populations sizes and timescales involved are amenable to human experimentation. But none of the models or experiments of which we are aware are genuinely open ended, in the sense of this paper.

The previous two paragraphs also apply, *mutatis mutandis*, to socio-economic systems. One could argue that the emergence of human socio-political systems is the latest manifestation of the emergence of eusociality in primates, and as such is a subset of biological evolution. Hence, it should be largely appropriate to apply a general model of perpetual emergence of novelty in biological evolution to socio-economic systems—but at different temporal and physical scales. It would be difficult to test such models experimentally, since we lack both the time and resources for controlled experimentation on entire human social systems, and it would be unethical in any case. However, historical data provide an empirical basis for model testing in this domain. A sufficiently rich model of socio-economic evolution should predict the emergence of new social, political, or economic forms of organization,

consistent with historical experience [see Morris (2010) for attempts in that direction].

More to the point of this paper, biological and socio-economic evolution are both subject to the same general theory of open-endedness, and so a sufficiently good model should apply to both. A general theory should provide a dictionary that would help us translate results from one domain into the other. One could never state that human social structures are the same as, say, bacterial biofilms. But a unifying theory would allow one to express exactly what they have in common, without simply anthropomorphising bacteria or dehumanizing people.

By the same argument, computational models based on a general theory of open-endedness should share enough with biological or socio-economic models to lay out exactly how computer simulations of open-endedness inform hypotheses about the emergence of novelty in biology and society. There is an established field of artificial society simulation, and simulations of biological phenomena are legion. Few of these computer simulations, however, explicitly present a model for the emergence of novelty together with a theoretical argument that the model is the same for both the simulation and the system being simulated.

We have argued that such a general theory of open-endedness is possible, and that it can lead to meta-models of the emergence of novelty that are appropriate for computer simulations. We do not claim, however, that the specific meta-model we present is always the most appropriate for a given analysis. Nor have we demonstrated how our computer simulation model is appropriate beyond the domain of computer simulations. There is probably no single meta-model for all questions, even if one has a general theory of open-endedness. Nevertheless, our hope is that attempting to develop such a theory, while remaining cognizant of the specific models that it would support, can be useful for prediction and experimentation across multiple domains.

There remain both conceptual and technical barriers to developing such a theory. In a sense, these are *grand challenges* to open-endedness. We state some of these challenges more explicitly using the terminology we developed above.

C-1: It remains an open question how to fully characterize how specific structuring rules (shortcuts) constrain specific classes of open-endedness. Consequently, it is unclear how to design a sufficiently rich set of generative rules such that no structuring rules are needed to explore even a three-level model. Thus, we challenge others and us to develop a simulation that can: (1) "write" its own structuring rules that act as shortcuts to bootstrap itself to the next level; (2) learn the structuring rules that fix the downward causation arrows (Ellis 2011); (3) recognize higher levels and higher level entities.

C-2: The hierarchical meta-model we present above is largely mereological, which has limitations that need to be explored. In particular, when a higher level of abstraction affects a lower level, the system is no longer describable in terms of a strict hierarchy. For example, when an ecosystem, which is an aggregation of individuals, changes the abiotic resources available to individuals, individuals adapt, changing the ecosystem.

It is even unclear what an "individual" or level zero entity is, in natural systems. For example, a human is an aggregation of thousands of species of bacterial, human, and viral genomes, where each constituent affects both the others and aggregations of the others. For example, endogenous retroviruses move genes into human genomes, where the genes reproduce and evolve in response to human behavior. Genes from endosymbiosis, such as those in the mitochondrial ancestors, move into the host genomes, where again they evolve according the activities of the host.

It may be the case that some specific research questions can *only* be answered with non-mereological models. So the challenge is to develop non-mereological models and demonstrate them in simulation.

C-3: Another challenge is to develop metrics of open-endedness that can be applied across the full range of biological, socio-economic, and computational complex systems. How open-ended does a model have to be to, say, predict when an endosymbiotic gene will migrate into a host, and then into a host of the host?

C-4: There are specific emergent systems that remain to be modeled and simulated. In particular, it would be helpful to have sufficiently rich models and meta-models, along the lines we have presented, with which the major transitions of life can be captured (Maynard Smith and Szathmáry 1995).

C-5: We also challenge others and us to model and simulate the movement of internal information systems such as genes, to external ones such as databases, and then back into internal ones, for example by genetic manipulation.

C-6: Another challenge is to model and simulate the emergence of high-level socio-economic entities such as corporations or governments, that then change the behavior of their constituents in such a way that new levels of socio-economic entities emerge, for example by merger or revolution.

C-7: A challenge in the realm of social systems would be the emergence of culture or specific languages, which then affect the nature of other cultures or languages.

Given the absence of a theory, we believe that novelty, innovation, emergence and open-endedness constitute some of the most fascinating and fruitful topics in the scientific discourse of today.

Acknowledgments We thank the anonymous referees for their insightful comments, which have helped to improve this paper from an earlier version. The authors acknowledge funding from diverse agencies: W. Banzhaf from NSERC under Discovery Grant RGPIN 283304-2012, B. Baumgaertner and J. A. Foster from the BEACON Center for Evolution in Action and from IBEST, G. Beslon and S. Stepney from the European Commission 7th Framework Programme (FPFP7-ICT-2013.9.6 FET Proactive: Evolving Living Technologies) EvoEvo project (ICT-610427), V.V. de Melo from Brazilian Government CNPq (Universal) grant (486950/2013-1) and Brazilian Government CAPES (Science without Borders Program) grant (12180-13-0), L. Spector from the National Science Foundation under Grant Nos. 1129139 and 1331283. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the CAPES/CNPq (Brazilian Government), European Commission (EU), NSERC (Canada), or NSF (USA). The authors are grateful to Memorial University for providing infrastructure for our workshop and to Overleaf for providing an excellent collaborative tool for writing a LaTeX document.

Appendix

On levels

Levels in philosophy

There are at least two different approaches one could take to characterizing the notion of level. On the one hand, philosophers like Wimsatt have provided a *prototype of levels* (as opposed to providing definitions) (Wimsatt 1994). Under this treatment, levels are distinguished by a cluster of rankable features: objects at different levels will have different *sizes*; objects at different levels stand in *composition* relations to one another; objects at the same level are governed by the same *laws* and the forces at play have similar magnitudes; objects at the same level are *reliable detectors* of one another because they stand in *regular* and *predictable* relations with one another; objects at the same level are investigated with the same set of *techniques* with respect to similar *disciplinary perspectives*. The advantage of providing a prototype treatment of levels is that some examples of levels may lack one or more of the aforementioned features.

On the other hand, philosophers like Craver have approached the issue by providing a taxonomy of the different *senses of level* (Craver 2007). This approach highlights the similarities and differences between the senses of level and can provide clarity where there are often misleading associations between them. For example, Craver distinguishes between four different senses of levels of composition: mereology, aggregativity, mere material/spatial containment, and mechanism:

1. Levels of mereology are formed by part–whole relations so that the collection of parts are at a lower level than the object that the parts make up. On the mereological conception of levels, complex things are regarded as wholes, but this does not emphasize whether the part–whole relation is constituted by material or spatial containment, or some other feature. It also does not specify what relations hold between the parts.
2. Levels of aggregativity, on the other hand, specify that the properties of items at a higher level are the simple sums of the properties of the items at the lower level. For example, the mass of a pile of sand is the sum of the masses of the individual grains.
3. Levels of mere material/spatial containment are permissive conception of composition. In this sense of level, an entity at a higher level is constituted by *pieces*. For example, to model climate we might divide the atmosphere into cubic-kilometer pieces. *Pieces* are to be contrasted with *components*. If we divide the human body into cubic-centimeter pieces, we would have a haphazard collection of things that do not have clear contributions to the workings of the human body.
4. Dividing the human body into components, however, involves the identification of how the part is relevant to the behavior of the whole. This is the defining feature of levels of mechanisms. On this conception of levels, components at a lower level are *organized together* to form components at a higher level, such that the behaviors of the components are relevant to the behavior of the whole.

An important aspect of thinking about mechanistic levels is that they are defined within a hierarchically organized mechanism; levels are not defined by objective kinds that are independently identifiable from a mechanism. To ask whether a given molecule and cell are at different levels makes sense only in the context of a given mechanism, which in turn makes sense only in a given model: the actual levels are model dependent. So, there is no unique answer to whether two items are at the same level; they are if they are both components in the same mechanism without being components of each other. However, under a different decomposition hierarchy, those items can be at different levels (for example, if one is a component of the other). Thus, an elephant and a bacterium it carries can be at the same level if they are components of the same mechanism, but can be at different levels if the hierarchical mechanism in question is the elephant and the components that make it up include bacteria. In recent years, level identification has also become an interesting topic of mathematical and information theoretic analysis (Pfante 2014).

On timescales and levels

Levels and timescales, both of dynamics and evolution, may be intimately related, but the relationship might be

complicated. In abiotic systems it may be the case that dynamics are faster at lower than at higher levels, but this may be a relatively trivial observation in that the higher levels are simply large-scale patterns of the lower level entities, and the patterns to be observed and said to exist must have a longer timescale. Thus, the large-scale weather system depends for its existence on the Brownian motions of the atmospheric molecules, whereas the process of self-organization generating the pattern requires many times the Brownian timescale to unfold. But here the weather system's timescale can only be defined in terms of the large-scale spatial pattern. Bénard cells provide another example. In this case, the lower level building blocks are much more stable than the emergent higher level structures but the low-level dynamic is much more rapid than the high-level dynamic.

In biological systems the relationship is more varied, and also more important. Living systems are not just compositions of molecules; they themselves construct most of their composite molecules. It is frequently claimed that the timescale of biological (Darwinian) evolution is slower than that of the individuals in the evolving population, so this is also a case of the higher level (species) having a slower timescale than the lower level (individuals or populations of individuals). It is also the case that the inherent rate of the appearance of new molecules in the genome is much greater than the rate of evolution in the individual, but in this case the evolution is largely suppressed by the repair mechanisms available to the phenotype to ensure the survival of the phenotype level, and hence also the survival of the genotype. As a result, the rate of evolution of the two levels is effectively the same—lower level variation uncorrelated with high-level variation simply dies out.

In human systems, the situation may be even less straightforward. A corporation may be considered to be a higher level entity in that it has a distinct physical existence (buildings, equipment, etc.) as well as a legal existence: incorporation gives it much the same legal status as a human being, and allows it to act in relevant respects as a person. Yet it is composed of a collection of people, who as employees become components of the corporate entity. The timescale of their evolution in terms of their function (for example, the products they make) may be slow relative to the lifetime of an employee, or much faster. Financial entities show an even greater range of evolutionary timescales, from a century or more in the case of savings banks to years or even months in the case of derivatives or closed-end funds. In the case of derivatives, the higher level evolves much faster than the lower level of the instruments being securitized. In human systems, we may even find circular hierarchies when individuals are subject to the constraints of various higher level organizations such as their employer, their church, and the bylaws of the local government, these organizations being in turn subject to the laws and regulations of the national government, but the national government being subject to a president/king/dictator—an individual belonging to the lowest level of the hierarchy. Note that humans have had the ability to build entities—computers—faster than themselves that now drive large parts of, e.g., financial exchanges.

At this point it would seem safe to say that for biological and human systems the relationship between levels and timescales is one that is fundamentally important. However, it is difficult to identify simple generalizations. The problem is one that for the time being must be examined in particular contexts.

On computational limits

Classical theoretical computer science investigates the relationships between objects that can be described or instantiated with algorithms, which are formalizations of Hilbert's concept of an "effective mathematical procedure" (Hilbert 1901). Several different algorithmic models exist, the most well-known being grammars in formal languages (Post 1944), recursive functions (Church 1936), and finite automata (Turing 1936). Less well known, but relevant to our discussion, are automata that infer functions from examples (Gold 1967; Valiant 1984). The objects described or instantiated by these models include primarily sets of finite (but indefinitely long) strings over finite alphabets, known as *formal languages*, and functions which map either strings onto strings or natural numbers onto natural numbers.

The fundamental results of theoretical computer science prove that there are hierarchies of sets of objects described or instantiated by abstract models of algorithms, such that variations in the models determine which level of the hierarchy the model describes or instantiates. In our terms, this means that even abstract mathematical models of algorithms define classes of innovations that are achievable by varying the models. Each of these classes of innovations are open-ended in the sense that there is no final or ultimate model; a model can always be extended so as to make a new class of objects accessible.

Consider some examples from automata theory. Turing machines are finite automata with indefinitely extensible memory. The finitude of this model implies that there is a hierarchy of three classes of sets or functions that can be described or instantiated: those that can be computed, those that can be enumerated, and those which are beyond computation. Augmenting the Turing machine model with access to a hypothetical (not actually implementable) external set, known as an *oracle*, produces an infinite, strict hierarchy of classes beyond these three [known as the Kleene–Post Hierarchy (Rogers 1987; Soare 1987)], and

therefore open-ended innovation in our sense. Restricting the Turing machine model by limiting access to memory produces another strict hierarchy which corresponds very closely to a hierarchy of formal languages known as the Chomsky Hierarchy (Chomsky 1956): random access to memory yields the class of languages describable by context-sensitive grammars, access to only the last item remembered yields the context-free grammars, and no access to memory yields the regular languages. Again, expanding a grammar with new types of production rules changes the level of the formal language hierarchy, moving up a well-defined hierarchy.

The branch of theoretical computer science known as *computational complexity* proves that restricting the number of steps or the amount of memory an automaton can use also produces strict hierarchies of formal languages (Hartmanis and Stearns 1965). Thus, for example, Turing machines that can run for a number of steps bounded by an exponential function of the input size can recognize languages that such machines bounded by a polynomial number of steps cannot. Similarly, automata that can access no more memory locations than some exponential of the input size can recognize languages that such automata with a polynomial bound cannot. Many of the most important problems in computer science involve understanding when different resource bounds or different types of automata (such as deterministic versus nondeterministic) produce genuine innovation, in the sense that one variation can recognize languages that the other cannot. For example, the famous *P versus NP* problem asks whether adding nondeterminism to polynomially bounded Turing machines is an innovation or not. If it is, then the hundreds of critically important practical problems in *NP* cannot be solved by algorithms that run in a reasonable amount of time (Garey and Johnson 1979; Cook 1971; Karp 1972). Unfortunately, we do not know the answers to most of these questions.

In summary, there are three lessons to learn from theoretical computer science that are relevant to this paper. First, enumeration can indeed provide innovation, since algorithms exist to enumerate some sets that no algorithm can fully describe. But this fact requires the abstraction of what an algorithm can do "in the limit", including the possibility that the algorithm never halts. Which brings us to the second lesson: computational complexity theory does indeed provide a context in which one can prove that certain variations in finite computations, such as allowing exponentially more memory access, generate innovations. Unfortunately, this theory breaks down precisely where it becomes useful: with reasonable limits on the number of steps or the amount of memory that a computation requires. Therefore, theoretical computer science is a useful framework in which to study open-ended innovation, but only in an abstract, mathematical sense. In particular, it does not currently directly explain open-ended innovation in physical systems such as computer simulations or biology.

References

Adami C, Brown CT (1994) Evolutionary learning in the 2D artificial life system avida. In: Artificial life IV: proceedings of the 4th international workshop on the synthesis and simulation of living systems, pp 377–381

Amar P, Legent G, Thellier M, Ripoll C, Bernot G, Nystrom T, Saier M, Norris V (2008) A stochastic automaton shows how enzyme assemblies may contribute to metabolic efficiency. BMC Syst Biol 2:27

Andrews PS, Polack FAC, Sampson AT, Stepney S, Timmis J (2010) The CoSMoS process, version 0.1: a process for the modelling and simulation of complex systems. Technical Report YCS-2010-453, Department of Computer Science, University of York

Andrews PS, Stepney S, Hoverd T, Polack FA, Sampson AT, Timmis J (2011) CoSMoS process, models, and metamodels. In: CoSMoS 2011: proceedings of the 2011 workshop on complex systems modelling and simulation, pp 1–14

Andrews PS, Stepney S, Timmis J (2012) Simulation as a scientific instrument. In: Proceedings of the 2012 workshop on complex systems modelling and simulation, Orleans, France, pp 1–10

Anderson PW (1972) More is different. Sci 177:393–396

Banzhaf W, Nordin P, Keller R, Francone F (1998) Genetic programming: an introduction. Morgan Kaufmann, San Francisco, CA, USA

Banzhaf W, Yamamoto L (2015) Artificial chemistries. MIT Press, Cambridge, MA, USA

Baptista T, Costa E (2013) Step evolution: improving the performance of open-ended evolution simulations. In: IEEE symposium on artificial life, Singapore 2013, pp 52–59. IEEE

Barricelli NA (1962) Numerical testing of evolution theories: part i theoretical introduction and basic tests. Acta Biotheor 16(1–2):69–98

Barrick JE, Lenski RE (2013) Genome dynamics during experimental evolution. Nat Rev Genet 14:827–839

Barrick JE, Yu DS, Yoon SH, Jeong H, Oh TK, Schneider D, Lenski RE, Kim JF (2009) Genome evolution and adaptation in a long-term experiment with *Escherichia coli*. Nature 461:1243–1247

Batut B, Knibbe C, Marais G, Daubin V (2014) Reductive genome evolution at both ends of bacterial population size spectrum. Nat Rev Microbiol 12(12):841–850

Baugh D (2015) Implementing von Neumann's architecture for machine self reproduction within the tierra artificial life platform to investigate evolvable genotype-phenotype mappings. PhD, Dublin City University. School of Electronic Engineering

Bedau MA (1991) Can biological teleology be naturalized? J Philos 88:647–655

Bedau MA (1996) The nature of life. In: Boden M (ed) The philosophy of artificial life. Oxford University Press, Oxford, UK, pp 332–357

Bedau MA (1999) Can unrealistic computer models illuminate theoretical biology? In: Proceedings of the 1999 genetic and evolutionary computation conference, workshop companion, pp 20–23

Bedau MA, Packard NH (1992) Measurement of evolutionary activity, teleology, and life. In: Langton C, Taylor C, Farme D, Rasmussen S (eds) Artificial life II. Addison-Wesley, Reading, MA, USA, pp 431–461

Bedau MA, Snyder E, Packard NH (1998) A classification of long-term evolutionary dynamics. In: ALife IV, MIT Press, Cambridge, MA, USA, pp 228–237

Bedau MA, McCaskill JS, Packard NH, Rasmussen S, Adami C, Green DG, Ikegami T, Kaneko K, Ray TS (2000) Open problems in artificial life. Artif Life 6(4):363–376

Bentley PJ (2003) Evolving fractal gene regulatory networks for robot control. In: ECAL 2003, vol 2801 of LNCS. Springer, Berlin, Germany, pp 753–762

Berlekamp ER, Conway JH, Guy RK (1982) Winning ways for your mathematical plays, volume 2: games in particular. Academic Press, New York, NY, USA

Bianco R, Nolfi S (2004) Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. Connect Sci 16(4):227–248

Blount ZD, Borland CZ, Lenski RE (2008) Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. Proc Natl Acad Sci USA 105:7899–7906

Buss L (1987) The evolution of individuality. Princeton University Press, Princeton, NJ, USA

Channon A (2001) Passing the ALife test: activity statistics classify evolution in Geb as unbounded. In: ECAL '01: proceedings of the 6th European conference on artificial life. Springer, Berlin, Germany, pp 417–426

Channon A (2003) Improving and still passing the ALife test: component-normalised activity statistics classify evolution in Geb as unbounded. In: Proceedings of artificial life VIII. MIT Press, Cambridge, MA, USA, pp 173–181

Channon AD, Damper RI (2000) Towards the evolutionary emergence of increasingly complex advantageous behaviours. Int J Syst Sci 31(7):843–860

Chomsky N (1956) Three models for the description of language. IRE Trans Inf Theory 2:113–124

Church A (1936) An unsolvable problem of elementary number theory. Am J Math 58:345–363

Cook SA (1971) On the complexity of theorem-proving procedures. In: Proceedings ACM symposium on theory of computing, pp 151–158

Craver C (2007) A field guide to levels. In: Explaining the brain: mechanisms and the Mosaic Unity of Neuroscience, chapter 5. Clarendon Press, Oxford

Dewdney AK (1987) Computer recreations: a program called mice nibbles its way to victory at the first core wars tournament. Sci Am 256(1):8–11

Dittrich P, Ziegler J, Banzhaf W (2001) Artificial chemistries: a review. Artif Life 7:225–275

Droop A, Hickinbotham S (2012) A quantitative measure of non-neutral evolutionary activity for systems that exhibit intrinsic fitness. Artificial Life XIII, pp 45–52

Edmonds B (1998) Meta-genetic programming: co-evolving the operators of variation. CPM Report 98–32, Centre for Policy Modelling, Manchester Metropolitan University, UK, Aytoun St., Manchester, M1 3GH, UK

Ellis GFR (2011) Top-down causation and emergence: some comments on mechanisms. R Soc Interf Focus 6(1):1–15

Fernández JD, Lobo D, Martn GM, Doursat R, Vico FJ (2012) Emergent diversity in an open-ended evolving virtual community. Artif Life 18(2):199–222

Fernando C, Kampis G, Szathmáry E (2011) Evolvability of natural and artificial systems. Proc Comput Sci 7:73–76

Foster JA (2001) Computational genetics: evolutionary computation. Nat Rev Genet 2:428–436

Frigg R, Hartmann S (2012) Models in science. In: Zalta EN (ed) The Stanford encyclopedia of philosophy. Fall 2012 edition

Gardner M (1970) Mathematical games: the fantastic combinations of John Conway's new solitaire game "life". Sci Am 223(4):120–123

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman, San Francisco, CA, USA

Gold EM (1967) Language identification in the limit. Inform Contr 10:447–474

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA, USA

Gotts NM (2009) Ramifying feedback networks, cross-scale interactions, and emergent quasi individuals in Conway's Game of Life. Artif Life 15(3):351–375

Hartmanis J, Stearns RE (1965) On the computational complexity of algorithms. Trans Am Math Soc 117:285–306

Harvey I (1992) Species adaptation genetic algorithms: a basis for a continuing SAGA. In: Toward a practice of autonomous systems: proceedings of the 1st European conference on artificial life, pp 346–354

Hasegawa T (2015) On the evolution of genotype-phenotype mapping: exploring viability in the Avida articial life system. PhD, Dublin City University, School of Electronic Engineering

Heylighen F (2012) Brain in a vat cannot break out. J Conscious Stud

Hickinbotham S, Clark E, Nellis A, Stepney S, Clarke T, Young P (2016) Maximising the adjacent possible in automata chemistries. Artif Life 22(1):49–75

Hickinbotham S, Clark E, Stepney S, Clarke T, Nellis A, Pay M, Young P. Specification of the stringmol chemical programming language version 0.2. Technical report, Technical Report YCS-2010-458, University of York

Hilbert D (1901) Mathematical problems. Archiv der Mathematik und Physik, 3rd series, vol 1, pp 44–65, 213–237

Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI, USA

Horsman C, Stepney S, Wagner RC, Kendon V (2014) When does a physical system compute? Proc R Soc A 470(2169):20140182

Hoverd T, Stepney S (2011) Energy as a driver of diversity in open-ended evolution. In: ECAL 2011, Paris, France, August 2011. MIT Press, Cambridge, MA, USA, pp 356–363

Humphreys P (2004) Extending ourselves: computational science, empiricism, and scientific method. Oxford University Press, Oxford, UK

Huneman P (2012) Determinism, predictability and open-ended evolution: lessons from computational emergence. Synthese 185(2):195–214

Hutton TJ (2002) Evolvable self-replicating molecules in an artificial chemistry. Artif Life 8(4):341–356

Kaneko K (1994) Chaos as a source of complexity and diversity in evolution. Artif Life 1(1/2):163–178

Kantschik W, Dittrich P, Brameier M, Banzhaf W (1999) Meta-evolution in graph GP. In: Poli R, Nordin P, Langdon W, Fogarty T (eds) Genetic programming: proceedings EuroGP 1999. Springer, Berlin, Germany, pp 15–28

Karp RM (1972) Reducibility among combinatorial problems. In: Miller RE, Thatcher JW (eds) Complexity of computer computations. Plenum, New York

Kimura M (1984) The neutral theory of molecular evolution. Cambridge University Press, Cambridge, UK

Kleppe A, Warmer J, Bast W (2003) MDA explained: the model driven architecture: practice and promise. Addison-Wesley, Reading, MA, USA

Knibbe C, Coulon A, Mazet O, Fayard J-M, Beslon G (2007) A long-term evolutionary pressure on the amount of noncoding DNA. Mol Biol Evolut 24(10):2344–2353

Koestler A (1970) Beyond atomism and holism: the concept of the holon. Perspect Biol Med 13(2):131–154

Koza J (1992) Genetic programming. MIT Press, Cambridge, MA, USA
Lan D (2006) Hierarchy, complexity, society. In: Pumain D (ed) Hierarchy in natural and social sciences. Springer, Berlin, Germany, pp 81–119
Lehman J, Stanley KO (2011) Abandoning objectives: evolution through the search for novelty alone. Evol Comput 19(2):189–223
Lehman J, Stanley KO (2012) Beyond open-endedness: quantifying impressiveness. ALIFE XIII, pp 75–82
Maley C (1999) Four steps toward open-ended evolution. In: Proceedings of the genetic and evolutionary computation conference (GECCO-1999), vol 2, pp 1336
Markovitch O, Sorek D, Lui LT, Lancet D, Krasnogor N (2012) Is there an optimal level of open-endedness in prebiotic evolution? Orig Life Evolut Biosph 42(5):469–474
Maynard Smith J (1988) Evolutionary progress and levels of selection. In: Nitecki M (ed) Evolutionary progress. University of Chicago Press, Chicago, IL, USA, pp 219–230
Maynard Smith J, Szathmáry E (1995) The major transitions in evolution. Oxford University Press
McCutcheon JP, Moran NA (2012) Extreme genome reduction in symbiotic bacteria. Nat Rev Microbiol 10(1):13–26
McMullin B (2012) Architectures for self-reproduction: abstractions, realisations and a research program. In: Adami C, Bryson DM, Ofria C, Pennock RT (eds) Artificial life XIII. MIT Press, Cambridge, MA, USA, pp 83–90
Medernach D, Kowaliw T, Ryan C, Doursat R (2013) Long-term evolutionary dynamics in heterogeneous cellular automata. In: Proceedings of the 15th annual conference on Genetic and evolutionary computation (GECCO'13), pp 231–238. ACM
Morris I (2010) Why the west rules-for now: the patterns of history and what they reveal about the future. Profile Books, London, UK
Nehaniv CL, Hewitt J, Christianson B, Wernick P. What software evolution and biological evolution don't have in common. In: 2nd international IEEE workshop on software evolvability (SE'06), pp 58–65. IEEE
Odling-Smee F, Laland K, Feldman M (2003) Niche construction: the neglected process in evolution. Princeton University Press, Princeton, NJ, USA
Pargellis AN (2001) Digital life behavior in the amoeba world. Artif Life 7(1):63–75
Pfante O, Bertschinger N, Obrich E, Ay N, Jost J (2014) Comparison between different methods of level identification. Adv Complex Syst 17:1450007–1–1450007-21
Plucain J, Hindré T, Le Gac M, Tenaillon O, Cruveiller S, Médigue C, Leiby N, Harcombe WR, Marx CJ, Lenski RE, Schneider D (2014) Epistasis and allele specificity in the emergence of a stable polymorphism in *Escherichia coli*. Science 343:1366–1369
Popper K (1982) The open universe: an argument for indeterminism. Hutchinson, London, UK
Post EL (1944) Recursively enumerable sets of positive integers and their decision problems. Bull Am Math Soc 50:284–316
Rasmussen S, Chen L, Deamer D, Krakauer DC, Packard NH, Stadler PF, Bedau MA (2004) Transitions from nonliving to living matter. Science 303(5660):963–965
Rasmussen S, Knudsen C, Feldberg R, Hindsholm M (1990) The coreworld: emergence and evolution of cooperative structures in a computational chemistry. Physica 42D:111–134
Ray TS (1992) An approach to the synthesis of life. In: Langton CG, Taylor C, Farmer JD, Rasmussen S (eds) Artifical life II. Addison-Wesley, Reading, MA, USA, pp 371–408
Ray TS (1992) Evolution, ecology and optimization of digital organisms. Working paper 92–08-042, Santa Fe
Rendell P (2002) Turing universality of the game of life. In: Adamatzky A (ed) Collision-based computing. Springer, Berlin, Germany
Renner G, Ekárt A (2003) Genetic algorithms in computer aided design. Comput Aided Design 35(8):709–726
Rensch B (1959) Evolution above the species level. Methuen, London
Reynolds CW (1987) Flocks, herds and schools: a distributed behavioral model. ACM SIGGRAPH Comput Graph 21(4):25–34
Rogers H (1987) Theory of recursive functions and effective computability. MIT Press, Cambridge, MA, USA
Rosen R (1991) Life itself: a comprehensive inquiry into the nature, origin, and fabrication of life. Columbia University Press, New York, NY, USA
Ruiz-Mirazo K, Moreno A (2012) Autonomy in evolution: from minimal to complex life. Synthese 185(1):21–52
Ruiz-Mirazo K, Pereto J, Moreno A (2004) A universal definition of life: autonomy and open-ended evolution. Origins Life Evolut Biosph 34(3):323–346
Ruiz-Mirazo K, Umerez J, Moreno A (2008) Enabling conditions for 'open-ended evolution'. Biol Philos 23(1):67–85
Schrödinger E (1944) What is life? The physical aspect of the living cell. Cambridge University Press, Cambridge, UK
Schulman R, Yurke B, Winfree E (2012) Robust self-replication of combinatorial information via crystal growth and scission. PNAS 109(17):6405–6410
Sipper M, Sanchez E, Mange D, Tomassini M, Pérez-Uribe A, Stauffer A (1997) A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems. IEEE Trans Evolut Comput 1(1):83–97
Sipper M, Sanchez E, Mange D, Tomassini M, Pérez-Uribe A, Stauffer A (1998) An introduction to bioinspired machines. In: Bio-inspired computing machines towards novel computational architectures. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, pp 1–12
Skusa A, Bedau MA (2002) Towards a comparison of evolutionary creativity in biological and cultural evolution. In: ALife VIII. MIT Press, Cambridge, MA, USA, pp 233–242
Soare RI (1987) Recursively enumerable sets and degrees: a study of computable functions and computably generated sets. MIT Press
Soros LB, Stanley KO (2014) Identifying necessary conditions for open-ended evolution through the artificial life world of Chromaria. In: Artificial life 14: international conference on the synthesis and simulation of living systems, vol 14, pp 793–800
Spector L (2010) Towards practical autoconstructive evolution: self-evolution of problem-solving genetic programming systems. In: Riolo R, McConaghy T, Vladislavleva E (eds) Genetic programming theory and practice VIII, volume 8 of genetic and evolutionary computation, chapter 2, pp 17–33. Springer, Ann Arbor, USA, 20–22 May 2010
Spector L, Robinson A (2002) Genetic programming and autoconstructive evolution with the push programming language. Genet Program Evol Mach 3:7–40
Standish RK (2003) Open-ended artificial evolution. Int J Comput Intell Appl 3(2):167–175
Stepney S (2012) A pattern language for scientific simulations. In: Proceedings of the 2012 workshop on complex systems modelling and simulation, Orleans, France, pp 77–103
Stepney S, Alden K, Paul JLB, Andrews S, Droop A, Ghetiu T, Hoverd T, Polack FAC, Read M, Ritson CG, Sampson AT, Timmis J, Welch PH, Winfield AFT (2016) Engineering simulations as scientific instruments. Springer (in preparation), Berlin, Germany
Stepney S, Andrews PS (2015) CoSMoS special issue editorial. Nat Comput 14:1–6

Stepney S, Hoverd T (2011) Reflecting on open-ended evolution. In: ECAL '11: proceedings of the 11th European conference on artificial life. MIT Press, Cambridge, MA, USA, pp 781–788

Suppes P (1960) A comparison of the meaning and uses of models in mathematics and the empirical sciences. Synthese 12:287–301

Szathmáry E (2015) Toward major evolutionary transitions theory 2.0. Proc Natl Acad Sci 112(33):10104–10111

Taylor T (1999) From artificial evolution to artificial life. PhD thesis, The University of Edinburgh

Turing A (1936–1937) On computable numbers, with an application to the Entscheidungsproblem. Proc London Math Soc Ser 2 42:230–265

Valiant L (1984) A theory of the learnable. Commun ACM 27:1134–1135

Waddington C (2008) Paradigm for an evolutionary process. Biol Theory 3:258–266

Weisberg M (2013) Simulation and similarity: using models to understand the world. Oxford University Press, Oxford, UK

Wilson D (1997) Biological communities as functionally organized units. Ecology 78:2018–2024

Wimsatt W (1987) False models as means to truer theories. In: Nitecki N, Hoffman A (eds) Neutral models in biology. Oxford University Press, New York, pp 23–55

Wimsatt W (1994) The ontology of complex systems: levels, perspectives, and causal thickets. Can J Philos 20(Suppl):207–274

Winsberg E (2010) Science in the age of computer simulation. Chicago University Press, Chicago, IL, USA