

# Modelling and Implementing Open-Ended Evolutionary Systems

Patrik Christen

FHNW University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland  
patrik.christen@fhnw.ch

## Abstract

Having a model and being able to implement open-ended evolutionary systems is important for advancing our understanding of open-endedness. Complex systems science and newest generation high-level programming languages provide intriguing possibilities to do so, respectively. Here, some recent advances in modelling and implementing open-ended evolutionary systems are reviewed first. Then, the so-called allagmatic method to describe, model, implement, and interpret complex systems is introduced. After highlighting some current modelling and implementation challenges, model building blocks of open-ended evolutionary systems are identified, a system metamodel of open-ended evolution is formalised in the allagmatic method, and an implementation prototype with a high-level programming language is outlined. The proposed approach shows statistical characteristics of open-ended evolutionary systems and provides a promising starting point to interpret novelty generated at runtime.

## Introduction

The diversity and complexity of organisms created by biological evolution over the last billions of years is staggering. It is a never-ending story which invented all of nature so far and continues to add more and more inventions (Stanley, 2019; Stanley et al., 2017). Engineered physical systems, evolutionary and genetic algorithms, artificial intelligence, deep learning, and other computational methods are thus far not even close to the diversity, creativity, and open-endedness exhibited by biological evolution. The main deficiency is that all these computational systems reach an equilibrium state at some point from which they do not generate anything new anymore – they are essentially cul-de-sacs and this happens rather quickly.

Understanding the open-endedness of biological evolution is a grand challenge, considered one of the “millennium prize problems” (Bedau et al., 2000) in the field of artificial life. If implemented in an open-ended computational system, it would have major implications far beyond artificial life (Stanley, 2019; Stanley et al., 2017). It would allow us to invent virtually everything including new architectures, furniture, cars, games, and of course algorithms and software in general (Stanley, 2019; Stanley et al., 2017). It would

most likely bring us closer to strong artificial intelligence, since only biological evolution has created it so far (Stanley, 2019; Stanley et al., 2017).

Furthermore, open-endedness has been observed in various complex systems including human languages, legal systems, economic and financial systems, and technological innovation showing its relevance as well as urging its study (Bedau et al., 2019; Banzhaf et al., 2016). These systems are important for our society. Further understanding their open-ended dynamics where a system is completely reorganising itself from time to time (i.e. it crashes) is key to managing them. This feeds back onto some of our biggest challenges including climate change and socio-economic stability (Turner, 2020).

## Recent Advances in Modelling Open-Ended Evolutionary Systems

### Definitions

Although progress has been made, especially by the open-ended evolution community, much remains to be explored (Packard et al., 2019). Before having a closer look at modelling, we start with some preliminaries regarding the definition of open-ended evolution and open-endedness. *Open-endedness* has been defined as the ability to continually produce novelty and/or complexity whereby novelty is classified into variation, innovation, and emergence (Banzhaf et al., 2016). Based on creativity research (Boden, 2015), different terms for this classification were suggested, namely exploratory, expansive, and transformational novelty, respectively (Taylor, 2019). The latter terms will be used here to avoid interpretation issues with innovation and emergence. Regardless of the terminology, both definitions relate to a formal model and metamodel of the system under study. *Exploratory novelty* can be described using the current model, *expansive novelty* requires a change in that model but still uses concepts in the metamodel, and *transformational novelty* introduces new concepts necessitating a change in the metamodel (Banzhaf et al., 2016; Taylor, 2019). With their connection to model and metamodel, they provide a way to determine whether and which kind of nov-

elty emerges in an open-ended evolutionary system.

Defining *complexity* and its measurement in open-ended evolutionary systems is a topic of ongoing research too. Dolson et al. (2019) recommend an information-theoretic approach based on the count of informative sites across all components in a population and suggest to improve it by also accounting for all possible mutations and by considering epistatic interactions. Channon (2019) defines individual complexity as the diversity of adaptive components in the individual, i.e. the number of active genes. Furthermore, in evolutionary biology, information is quantified with respect to different sources available to an adapting organism, in particular from ancestors and the environment (Rivoire, 2016).

### **Modelling Contributions from Artificial Life and Open-Ended Evolution Community**

Banzhaf et al. (2016) have argued that open-endedness in physical systems such as in a computation are hard to prove in a finite universe and therefore one might look for producing a sufficient rather than an infinite number of open-ended events, which is then called *effectively* open-ended. To achieve this despite the limits on computational power, it has been suggested to hard-code certain elements of the model, e.g. the process of replication, into so-called *shortcuts* (Banzhaf et al., 2016; Taylor, 2019). Taylor (2019) bases shortcuts on generally accepted processes of Darwinian evolution: phenotype generation (from the genotype), phenotype evaluation, and reproduction with variation. Ongoing evolutionary activity and with that exploratory open-endedness is promoted by modifying the adaptive landscape, the topology of genetic space, or the genotype-phenotype map. He further argues that none of these expand the phenotype-space itself and thus do not help us for expansive and transformational open-endedness, where so-called *door-opening* states in phenotype-space are needed. The complexity of physical and chemical laws provides a vast space for biological systems whereas in computational systems one might dynamically increase the space instead, e.g. providing access to additional resources on the Internet (Boden, 2015; Taylor, 2019; Taylor et al., 2016). In contrast, at the third workshop on open-ended evolution, Taylor and others from the open-ended evolution community mentioned that current computational systems implement rather scanty environments and organisms.

Taylor (2019) also proposes two possible intrinsic mechanisms to access new states. The first is via *exaptation*, where a trait changes its function to a different one from the one it was originally adapted for. Physical systems are composed of multi-property components having multiple properties in different domains (mechanical, chemical, electrical, pressure, etc.) (Taylor, 2019). E.g. a multifunctional enzyme has multiple properties in the same domain, which can produce expansive novelties whereas transformational novelties

can be achieved by properties in different domains (Taylor, 2019). The second is via *non-additive composition*, which is phenotype generation by assembling a number of components drawn from a set of component types (Taylor, 2019). E.g. the construction of proteins from amino acid sequences, producing new molecules, introducing new functions (Taylor, 2019).

This assembly of lower-level elements into higher-level structures is also highlighted by Banzhaf et al. (2016). With the mentioned metamodel to define novelties, they also provide an abstract way to model multiple levels accounting for such constructed structures at different levels (Banzhaf et al., 2016). They also mention that having several levels drastically increases the combinatorial possibilities to construct new structures and with that also the demand for computational power (Banzhaf et al., 2016). It therefore seems to be a way to increase the opportunities to create something new. It also implies that open-ended evolution in computational systems is computational intensive.

### **Modelling Contributions from Complex Systems Science**

There are also relevant modelling contributions from the field of complex systems science. W. Brian Arthur is known for his work on complexity economics (Arthur, 2014) and technology evolution (Arthur, 2018, 2009a). He proposed the concept of *combinatorial evolution*, which states that new technologies are created out of existing technologies and iteratively, these newly created technologies become building blocks for yet further technologies (Arthur, 2009a,b). Technology is therefore self-creating or autopoietic (Arthur, 2018, 2009b). In a simple computer model of circuits, Arthur and Polak (2006) showed that complicated technologies (in their case circuits) could be created out of simpler building blocks and they found evidence of self-organised criticality. It requires some kind of modularity and the evolution of simpler stepping stone technologies (Arthur, 2018, 2009a,b; Arthur and Polak, 2006). The latter means that we cannot create a technology ahead of time without first creating the simpler precursor technologies. Natural phenomena also provide technological elements which can be combined (Arthur, 2018, 2009b). In terms of open-endedness, there seems to be a vast space of possible combinations and with the conversion of discovered natural phenomena into technological elements, there is a mechanism in place to expand that space.

Combinatorial evolution is also part of a more general approach to modelling evolution by the complex system scientist Stefan Thurner. He and his colleagues recently introduced the co-evolutionary, combinatorial, and critical evolution model (CCC model) (Thurner, 2018; Thurner et al., 2018, 2010; Klimek et al., 2012, 2010; Hanel et al., 2005). It models evolution as an open-ended process of creation and destruction of new entities emerging from the interac-

tions of existing entities with each other and with their environment (Thurner et al., 2018). The spaces of entities and of interactions co-evolve and new entities emerge spontaneously or through the combination of existing entities. This leads to power law statistics in histories of events (Thurner et al., 2018, 2010). Selection is modelled by specifying rules for what can be created and what will be destroyed (Thurner, 2018; Thurner et al., 2018, 2010). This model captures so-called *punctuated equilibria* in biological evolution (Gould and Eldredge, 1977) or *Schumpeterian business cycles* in economic evolution (Schumpeter, 1939), where an equilibrium is destabilised or destroyed by a critical transition leading to another equilibrium in an ongoing and thus open-ended process (Thurner, 2018; Thurner et al., 2018, 2010; Thurner, 2011). It is interesting to note that it could be shown that in economic innovation, *creative deconstruction* is happening and not *niche filling* as in biological innovation (Thurner, 2018; Thurner et al., 2018; Klimek et al., 2012).

### **Modelling Contributions from Artificial Intelligence and Evolutionary Algorithms**

Open-ended evolution is also studied in artificial intelligence and evolutionary algorithms. It is an emerging topic where the research of Kenneth O. Stanley serves as an example here. He tried to get rid of the prevailing concept to reward optimising a fitness function and has even suggested to abandon objectives in general (Stanley and Lehman, 2015; Lehman and Stanley, 2011; Stanley, 2010). He showed that a novelty-driven approach finds solutions faster and results in solutions with less genomic complexity in comparison to traditional evolutionary computation (Woolley and Stanley, 2014). He also devised a number of algorithms including novelty search with explicit novelty pressure, MAP-Elites and innovation engines with explicit elitism within niches in an otherwise divergent process, and minimal criterion co-evolution where problems and solutions can co-evolve divergently (Auger et al., 2019; Bosman et al., 2017). Similar to Thurner, avoiding objectives also allowed Stanley to model punctuated equilibria with transitions between equilibria in a simple simulation with voxel structures (Pugh et al., 2017). Also in this case co-evolution and the never-ending creation of anything new by combining existing structures were essential ingredients.

### **Modelling Contributions from Evolutionary Biology**

The work of Thurner and Stanley indicates that transitions between equilibria are an important part of open-ended evolution. In evolutionary biology, the major evolutionary transitions are of great interest too, for example the transition from unicellular to multicellular organisms (Szathmary, 2015; Szathmary and Smith, 1995). Here, only a small selection of research is presented, mainly on mechanisms which can explain rapid increases in diversity and biologi-

cal innovation. The work of evolutionary ecologist Ole Seehausen illustrates this well as he is interested in mechanisms by which diversity arises. Especially relevant here is the possibility of speciation through combinatorial mechanisms. In such cases, new combinations of old gene variants can quickly generate reproductively isolated species and thus provide a possible explanation for rapid speciation (Marques et al., 2019). E.g. he showed that hybridisation between two divergent lineages provides ample genetic starting variation. This is then combined and sorted into many new species fuelling rapid cichlid fish adaptive radiations (Meier et al., 2017). Seehausen furthermore investigates and underlines the importance of jointly considering species traits and environmental factors in speciation and adaptive radiation as they affect one another (Seehausen, 2009; Wagner et al., 2012). His work therefore supports the importance of co-evolutionary and combinatorial dynamics for open-ended evolution, even though co-evolution is between species in a heterogeneous environment and combinations happen at the gene level.

Biological insights into innovation itself are also relevant. The work of evolutionary biologist Andreas Wagner illustrates this nicely (Hochberg et al., 2017; Wagner, 2011b). E.g. he showed that recombination creates phenotypic innovation in metabolic networks much more readily than random changes in chemical reactions (Hosseini et al., 2016). The work of Wagner suggests that recombination of genetic material is a general mechanism which greatly increases the diversity of genotypes (Wagner, 2011a; Martin and Wagner, 2009). Also relevant here is his work on evolutionary innovation through exaptation. He found that simulated real metabolic networks were not only able to metabolise on a specific carbon source but also on several others, which shows that metabolic systems may harbour hidden pre-adaptations that could potentially lead to evolutionary innovations (Barve and Wagner, 2013). Combinatorial interactions at the gene-level again play a crucial role and the latter study revealing hidden pre-adaptations is similar to Stanley’s open-ended algorithms creating many potential solutions before applied to solve an actual problem.

Besides this limited and biased review of contributions from evolutionary biology, it seems nevertheless important to point out that the field has shown that combinatorial interactions matter at organisational levels above the genes and that a changing environment can greatly affect species diversity and vice versa.

## **Recent Advances in Implementing Open-Ended Evolutionary Systems**

### **Implementation Contributions from Artificial Life and Open-Ended Evolution Community**

We first consider implementations from the artificial life and open-ended evolution community. Banzhaf et al. (2016) and

Taylor (2019) provide some implementation suggestions. The implementation of computational systems which can detect and integrate novelties into the model and metamodel as described by Banzhaf et al. (2016) and Taylor (2019) provides a challenge in its own right. It is argued that operations should be defined *intrinsically* in the system and by the system itself (Taylor, 2019; Packard, 1988). It requires program code which can recognise and modify itself. Banzhaf et al. (2016) state that this can be achieved by representing entities as strings of assembly language code, or by using a high-level language specifically designed for this purpose (Spector and Robinson, 2002), or a *reflective* language. Indeed it has been possible to generate exploratory, expansive, and transformative novelty with Stringmol, where modifications happen in sequences of assembly language code (Stepney and Hickinbotham, 2020). A replicator and some of the observed operations and structures were extrinsically defined whereas some others could be defined intrinsically (Stepney and Hickinbotham, 2020).

There are a number of computational systems of which Avida (Ofria and Wilke, 2004) and Geb (Channon and Damper, 2010) are two prominent examples. Usually digital organisms are represented by assembly code competing for limited CPU resources. Most of these systems extrinsically implement common shortcuts such as replication and a certain fitness function, which makes them a powerful tool to explore biological questions such as the genotype-phenotype mapping (Fortuna et al., 2017) in a highly controlled way. Another strength of computational systems is that they usually involve visualisations, e.g. Sims (1994), and with that help exploring complex evolutionary dynamics. With respect to open-endedness, however, Pugh et al. (2017) point out that none of these systems has generated explosions of complexity, as seen in biological evolution during transitions and therefore something must still be missing. With Voxelbuild, Pugh et al. (2017) contributed the most relevant computational system in this respect. A first prototype demonstrated that a certain organisation of voxels emerged which was used as a stepping stone for yet other organisations appearing later. This seems to be similar to combinatorial evolution. Additionally, they report that exaptation occurred, which reminds of the evolutionary biology studies.

### **Implementation Contributions from Complex System Science**

Thurner et al. (2018) add another important aspect to the implementation. They argue that only a so-called *algorithmic* implementation and thus discrete formulation can work because in evolutionary systems, boundary conditions cannot be fixed (the environment evolves as a consequence of the system dynamics), and the phase-space is not well defined as it changes over time (Thurner et al., 2018). It would lead to a system of dynamical equations that are coupled dynamically to their boundary conditions, which is according to

them a mathematical monster and the reason why evolutionary systems cannot be implemented following an analytical approach. In addition, with the CCC model, they provide a general description of a complex evolving system that is so general that it applies to every evolutionary system.

## **The Allagmatic Method**

### **Modelling Contribution**

We have developed the so-called *allagmatic method* (Christen and Fabbro, 2019, 2020) to describe, model, implement, and interpret complex systems. It consists of a system metamodel inspired and guided by philosophical concepts of the French philosopher Gilbert Simondon (Simondon, 2017). His metaphysics gives an operational and systemic account of how technical and natural objects function. It allows abstractly defining a system with the concepts *structure* and *operation* since according to him, systems develop starting with a seed by a constant interplay between operations and structures. More concretely but still general, we defined model building blocks in a system metamodel. The main building blocks are *entity*, *milieu*, *update function*, *adaptation function*, and *target*, for which we recently provided a mathematical formalism (Christen, 2020). The creation of such a system model and metamodel can be followed through three regimes: In the virtual regime, abstract definitions with classes corresponding to interpretable philosophical concepts and principles are given. Using generic programming, the type of the states an entity can have are defined by defining a system model object with no other parameters initialised yet. Here starts the metastable regime, where step by step the object/model is concretised with parameters such as number of entities and concrete update functions (model individuation). Once all parameters are defined, the object can be executed in the actual regime. If there are any adaptation processes involved, the allagmatic method cycles between the metastable and actual regimes.

### **Implementation Contribution**

The programming of the allagmatic method with its system metamodel is aligned with philosophical concepts. This not only allows interpretation of the final result in the context of the related metaphysics, it also allows to follow the developmental steps a model is undergoing and thus provides a way to study the emergence of typical characteristics of complex systems. We recently outlined how adaptation can be studied in this way in a working paper (Fabbro and Christen, 2020). There, we also introduced the possibility and principles to form hierarchies and define control, which further supports the use of the allagmatic method to define concepts or principles that are difficult to pin down.

Furthermore, we showed how the method might be used for automatic programming (Christen and Fabbro, 2020). We found that the abstract model building blocks are well suited to be automatically combined by self-modifying code

in a high-level language. Our work shows that certain philosophical principles and even metaphysics as a whole can be defined and implemented in program code providing the opportunity to run these principles or the whole metaphysics and study them in action.

We also created a prototype of open-ended automatic programming by combinatorial evolution (Fix et al., 2021). Similar to Arthur and Polak (2006), we created a computational model based on combinatorial evolution but instead of evolving circuits, we evolved computer code. Useful code blocks were stored in a repository and could be used in later iterations. Starting with basic keywords available in the programming language, more complex code blocks including classes, void methods, and variable declarations evolved.

## **Current Modelling and Implementation Challenges**

### **Modelling Open-Ended Evolutionary Systems**

*Co-evolutionary dynamics, combinatorial interactions, and a changing environment* seem to be important ingredients of open-ended evolutionary systems. The work of evolutionary biologists including Seehausen and Wagner supports the view that co-evolutionary dynamics and combinatorial interactions are key elements. They also indicate that biological evolution exhibits different levels or types of combinatorial interactions and that the environment is an important driver and mediator of change. The CCC model accounts for co-evolutionary dynamics and combinatorial interactions, and successfully generates the statistics of economic data with ever reoccurring transitions between equilibria (Thurner, 2018; Thurner et al., 2018, 2010; Klimek et al., 2012, 2010; Hanel et al., 2005). It could also show that economic innovations are driven by creative destruction, thus Schumpeterian evolution. This provides important insights into open-ended dynamics of economic evolution (Thurner, 2018; Thurner et al., 2018; Klimek et al., 2012). However, it still needs to be investigated in other evolutionary systems, especially in biological evolution. Banzhaf et al. (2016) and Taylor (2019) provide guidance for modelling open-ended evolution in general which might allow us to come up with a model that captures open-ended dynamics of any evolutionary system, including economic and biological systems.

### **Implementing Open-Ended Evolutionary Systems**

There is the challenge of an intrinsic implementation of open-ended evolution. The programming techniques already exist to do that, however, the real challenge is linking the structure and events in the implementation with interpretable concepts. To illustrate this problem, let us assume giving up shortcuts completely and letting the program overwrite the model and metamodel completely. Having no replicator or other prevailing concepts makes it hard to understand and see what is going on in the evolutionary simulation. This

problem was discussed at the third workshop on open-ended evolution. It is mostly uncharted territory needing much more research, including how to identify certain concepts and components from simulation data and how to implement such systems in a purely intrinsic manner, where generated novelties are meaningfully integrated into the model/meta-model by the evolving systems themselves.

Another challenge is the choice of digital organisms and environment. The CCC model (Thurner, 2018; Thurner et al., 2018, 2010; Klimek et al., 2012, 2010; Hanel et al., 2005) provides a mathematical formalism for theoretical considerations and ways to perform statistical analyses. Computational systems from artificial life and open-ended evolution community such as Voxelbuild usually come with powerful visualisations, however, they lack a mathematical formalism.

## **The Allagmatic Method for Open-Ended Evolutionary Systems**

### **Model Building Blocks of Open-Ended Evolutionary Systems**

Observing evolving systems like technology or the rain forest makes clear that not only entities evolve but also interactions among them. Co-evolution implies that species affect each other reciprocally. Since species are also part of the environment, co-evolution leads to a changing environment providing more possibilities of state changes. Also external environmental input can change and affect species and their interactions. Combinatorial interactions create new entities from existing entities (Arthur, 2018, 2009a; Thurner, 2018; Thurner et al., 2018). These newly created entities might be able to exploit different parts of the changing environment and with that might be able to fill niches arising. Chromaria (Soros, 2018) captures this to some degree as entities become part of the environment thus change the environment that interacts with further new entities. Changing the interactions between entities and between entities and their environment leads to complex cascades of changes potentially leading to disruptive changes in the system that can be regarded as novelties. Combinatorial interactions also lead to evolutionary changes and potential novelties, they combine existing entities to form new entities. This can be nicely observed in the evolution of technology (Arthur, 2018, 2009a). It is a way how transitions might be explained, e.g. from unicellular to multicellular organisms (Szathmary, 2015; Szathmary and Smith, 1995).

The main idea is therefore to capture co-evolutionary dynamics including with the environment and combinatorial interactions with the allagmatic method as given by the CCC model. The CCC model has been able to generate an ongoing evolutionary process with punctuated equilibria when in addition the lifetime of an entity was limited (Thurner, 2018; Thurner et al., 2018, 2010). It therefore showed the statistical behaviour of open-ended evolutionary systems. Here, the

CCC model is formalised within the allagmatic method to allow interpretation within the modelled metaphysics. The system metamodel of the allagmatic method and the CCC model both follow a complex systems perspective, which makes them compatible.

The model building blocks or general principles to be captured with the allagmatic method are specifically evolving entities, entity lifetime parameters, co-evolutionary operations of entities and environment, and combinatorial interactions.

### System Model and Metamodel Formalism of Open-Ended Evolution

The allagmatic method consists of a system metamodel for modelling systems in general and complex systems in particular (see Christen (2020) for detailed mathematical definitions). The system metamodel describes individual parts of a system as entities defined with an entity  $e$ -tuple  $\mathcal{E} = (\hat{e}_1, \hat{e}_2, \hat{e}_3, \dots, \hat{e}_e)$ , where  $\hat{e}_i \in Q$  with  $Q$  being the set of  $k$  possible entity states. Entity states are updated over time according to an update function  $\phi : Q^{m+1} \rightarrow Q$  with  $m$  being the number of neighbouring or linked entities. The update function  $\phi$  therefore describes how entities evolve over time dependent on neighbouring entities. Entities are thereby considered connected together in a network structure and defined with the milieu  $e$ -tuple  $\mathcal{M} = (\hat{\mathcal{M}}_1, \hat{\mathcal{M}}_2, \hat{\mathcal{M}}_3, \dots, \hat{\mathcal{M}}_e)$ , where  $\hat{\mathcal{M}}_i = (\hat{m}_1, \hat{m}_2, \hat{m}_3, \dots, \hat{m}_m)$  is the milieu of the  $i$ -th entity  $\hat{e}_i$  of  $\mathcal{E}$  consisting of  $m$  neighbours of  $\hat{e}_i$ . Over time, update function  $\phi$  and milieu  $\mathcal{M}$  might be changing as well, which is described with the adaptation function  $\psi$ .

We now extend the system metamodel with principles to model open-ended evolution as identified above. *Evolving entities*: The entity  $e$ -tuple  $\mathcal{E}$  captures evolving entities in the same way as the general evolution algorithm (Thurner et al., 2018) does with the state vector  $\sigma$ . The general evolution algorithm can be regarded as the metamodel from which the CCC model can be created (Thurner et al., 2018). *Co-evolutionary operations of entities and environment*: With the creation of new entities (novelty), also new possibilities for interactions emerge. This is key for open-endedness and is captured by the co-evolution of entities and their interaction in the general evolution algorithm (Thurner et al., 2018). Formally, the update equations of the entity state vector  $\sigma$  and the interaction tensors  $M$  are simultaneously updated over time. In the system metamodel of the allagmatic method, this is described with the adaptation function  $\psi$  that can be modelled in such a way that the update function  $\phi$  and milieu  $\mathcal{M}$  are updated simultaneously. This is a concretisation of the system metamodel into a metamodel of open-ended evolution. The environment is also part of co-evolution and described in the state vector  $\sigma$  in the general evolution algorithm (Thurner et al., 2018) and the entity  $e$ -tuple  $\mathcal{E}$  in the system metamodel. *Combinatorial interac-*

*tions*: In complex systems, interactions are of combinatorial nature consisting of rules determining how new entities can be formed out of existing entities. The creation and destruction of entities is encoded in rules that do not change with time. They can be regarded as physical or chemical laws determining which transformations and reactions are possible, respectively. Please note that this includes the typical evolutionary mechanisms of selection, competition, and reproduction. At runtime, models created from this metamodel make use only of a subset of these rules at any given time point, which is captured with so-called active productive/destructive rules. This is formally described with the function  $F$  in the general evolution algorithm (Thurner et al., 2018) and the update function  $\phi$  in the system metamodel. *Entity lifetime parameters*: Besides the creation of new entities through combinatorial interactions, entities can spontaneously appear, which would be similar to discovering a new law or element in nature. By introducing a decay rate  $\lambda$ , the CCC model did not freeze (Thurner et al., 2018). It thus plays an important role for open-ended evolution. In the system model, this parameter can be described as a further structure in the update function  $\phi$ .

### Implementation with the Allagmatic Method

An intrinsic implementation as suggested by Banzhaf et al. (2016); Taylor (2019) requires self-modifying program code and some way to add novelties to the model or metamodel. Interpreting these novelties in the context of a certain metaphysics will most likely require a high-level language with the capabilities to modify program code during runtime and reflect on it. C# provides these capabilities with the open-source Roslyn .NET compiler. The compiler platform provides dynamic code manipulation with expression trees and many other features including reflection as well as comprehensive code analysis. Expression trees can either be created from a string containing program code or they can be assembled using predefined classes. Instead of writing program code into a file and then compile it, expression trees are stored as an object and compiled in-memory at runtime.

In the allagmatic method, a general layer in the system metamodel that is not be modifiable by the code is suggested here. These are the model building blocks every complex evolutionary system requires. However, there is also a layer in the metamodel that is modifiable by the code. It consists of less general model building blocks that are basically more concrete instances of the general layer. With these different layers and controllable code self-modification, it will potentially be possible to link concepts defined in the metamodel to newly generated code improving interpretability.

It is interesting to note that certain models anticipate changes that might occur to them. In every evolutionary system new entities arise and other entities disappear, which will not only change how many entities there are but also their interactions with each other and the environment. The

CCC model is capable of accounting for such changes in the model through co-evolution of entity states and interactions. On this level, it therefore does not need self-modification of the code but generic programming of certain structures to dynamically adapt them to these changes.

## Discussion and Outlook

Based on recent advances, the model building blocks *evolving entities*, *entity lifetime parameters*, *co-evolutionary operations of entities and environment*, and *combinatorial interactions* are identified to characterise open-ended evolutionary systems. These principles led to punctuated equilibria in the CCC model (Thurner et al., 2018), which means that it never reaches an equilibrium state and thus can be regarded open-ended. This study provides a formal description of a system metamodel for open-ended evolution according to the CCC model thus also capable of generating punctuated equilibria.

The motivation for such a system metamodel is that it is part of the allagmatic method, which allows describing, modelling, implementing, and interpreting abstract concepts and principles, i.e. a certain metaphysics. Interpretation of concepts within a metaphysics provides a promising starting point to interpret novelty generated at runtime.

## Acknowledgements

This work was supported by the Hasler Foundation under Grant 21017.

## References

- Arthur, W. B. (2009a). *The Nature of Technology: What It Is and How It Evolves*. Free Press, New York.
- Arthur, W. B. (2009b). Where Darwin doesn't fit. . . . *New Scientist*, 203(2722):26–27.
- Arthur, W. B. (2014). *Complexity and the Economy*. Oxford University Press, New York.
- Arthur, W. B. (2018). How We Became Modern. In Sim, S. and Seet, B., editors, *Sydney Brenner's 10-on-10: The Chronicles of Evolution*. Wildtype Books.
- Arthur, W. B. and Polak, W. (2006). The evolution of technology within a simple computer model. *Complexity*, 11(5):23–31.
- Auger, A., Stütze, T., Wang, R., Lehman, J., Clune, J., and Stanley, K. O. (2019). POET: open-ended coevolution of environments and their optimized solutions. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 142–151.
- Banzhaf, W., Baumgaertner, B., Beslon, G., Doursat, R., Foster, J. A., McMullin, B., Melo, V. V. d., Miconi, T., Spector, L., Stepney, S., and White, R. (2016). Defining and simulating open-ended novelty: requirements, guidelines, and challenges. *Theory in Biosciences*, 135(3):131–161.
- Barve, A. and Wagner, A. (2013). A latent capacity for evolutionary innovation through exaptation in metabolic systems. *Nature*, 500(7461):203–206.
- Bedau, M. A., Gigliotti, N., Janssen, T., Kosik, A., Nambiar, A., and Packard, N. (2019). Open-Ended Technological Innovation. *Artificial Life*, 25(1):33–49.
- Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., and Ray, T. S. (2000). Open Problems in Artificial Life. *Artificial Life*, 6(4):363–376.
- Boden, M. A. (2015). Creativity and ALife. *Artificial Life*, 21(3):354–365.
- Bosman, P. A. N., Brant, J. C., and Stanley, K. O. (2017). Minimal criterion coevolution: a new approach to open-ended search. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 67–74.
- Channon, A. (2019). Maximum Individual Complexity is Indefinitely Scalable in Geb. *Artificial Life*, 25(2):134–144.
- Channon, A. D. and Damper, R. I. (2010). Towards the evolutionary emergence of increasingly complex advantageous behaviours. *International Journal of Systems Science*, 31(7):843–860.
- Christen, P. (2020). Model Creation and Equivalence Proofs of Cellular Automata and Artificial Neural Networks. *arXiv*. arXiv:2005.01192.
- Christen, P. and Fabbro, O. D. (2019). Cybernetical Concepts for Cellular Automaton and Artificial Neural Network Modelling and Implementation. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 00:4124–4130. arXiv:2001.02037.
- Christen, P. and Fabbro, O. D. (2020). Automatic Programming of Cellular Automata and Artificial Neural Networks Guided by Philosophy. In Dornberger, R., editor, *New Trends in Business Information Systems and Technology*, number 294 in Studies in Systems, Decision and Control, pages 131–146. Springer, Cham. arXiv:1905.04232.
- Dolson, E. L., Vostinar, A. E., Wiser, M. J., and Ofria, C. (2019). The MODES Toolbox: Measurements of Open-Ended Dynamics in Evolving Systems. *Artificial Life*, 25(1):50–73.
- Fabbro, O. D. and Christen, P. (2020). Individuation and Adaptation in Complex Systems. *arXiv*. arXiv:2009.00110.
- Fix, S., Probst, T., Ruggli, O., Hanne, T., and Christen, P. (2021). Open-Ended Automatic Programming by Combinatorial Evolution. *arXiv*.
- Fortuna, M. A., Zaman, L., Ofria, C., and Wagner, A. (2017). The genotype-phenotype map of an evolving digital organism. *PLOS Computational Biology*, 13(2):e1005414.
- Gould, S. J. and Eldredge, N. (1977). Punctuated equilibria: the tempo and mode of evolution reconsidered. *Paleobiology*, 3(2):115–151.
- Hanel, R., Kauffman, S. A., and Thurner, S. (2005). Phase transition in random catalytic networks. *Physical Review E*, 72(3):036117.
- Hochberg, M. E., Marquet, P. A., Boyd, R., and Wagner, A. (2017). Innovation: an emerging focus from cells to societies. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1735):20160414.

- Hosseini, S.-R., Martin, O. C., and Wagner, A. (2016). Phenotypic innovation through recombination in genome-scale metabolic networks. *Proceedings of the Royal Society B: Biological Sciences*, 283(1839):20161536.
- Klimek, P., Hausmann, R., and Thurner, S. (2012). Empirical Confirmation of Creative Destruction from World Trade Data. *PLoS ONE*, 7(6):e38924.
- Klimek, P., Thurner, S., and Hanel, R. (2010). Evolutionary dynamics from a variational principle. *Physical Review E*, 82(1):011901.
- Lehman, J. and Stanley, K. O. (2011). Abandoning Objectives: Evolution Through the Search for Novelty Alone. *Evolutionary Computation*, 19(2):189–223.
- Marques, D. A., Meier, J. I., and Seehausen, O. (2019). A Combinatorial View on Speciation and Adaptive Radiation. *Trends in Ecology & Evolution*, 34(6):531–544.
- Martin, O. C. and Wagner, A. (2009). Effects of Recombination on Complex Regulatory Circuits. *Genetics*, 183(2):673–684.
- Meier, J. I., Marques, D. A., Mwaiko, S., Wagner, C. E., Excoffier, L., and Seehausen, O. (2017). Ancient hybridization fuels rapid cichlid fish adaptive radiations. *Nature Communications*, 8(1):14363.
- Ofria, C. and Wilke, C. O. (2004). Avida: A Software Platform for Research in Computational Evolutionary Biology. *Artificial Life*, 10(2):191–229.
- Packard, N., Bedau, M. A., Channon, A., Ikegami, T., Rasmussen, S., Stanley, K. O., and Taylor, T. (2019). An Overview of Open-Ended Evolution: Editorial Introduction to the Open-Ended Evolution II Special Issue. *Artificial Life*, 25(2):93–103.
- Packard, N. H. (1988). Intrinsic adaptation in a simple model for evolution. In Langton, C. G., editor, *Artificial Life*. Addison-Wesley, Reading.
- Pugh, J. K., Soros, L. B., Frota, R., Negy, K., and Stanley, K. O. (2017). Major evolutionary transitions in the Voxelbuild virtual sandbox game. *The Fourteenth European Conference on Artificial Life*, pages 553–560.
- Rivoire, O. (2016). Informations in Models of Evolutionary Dynamics. *Journal of Statistical Physics*, 162(5):1324–1352.
- Schumpeter, J. A. (1939). *Business Cycles*. McGraw-Hill, London.
- Seehausen, O. (2009). Speciation affects ecosystems. *Nature*, 458(7242):1122–1123.
- Simondon, G. (2017). *On the Mode of Existence of Technical Objects*. University of Minnesota Press, Minneapolis/London.
- Sims, K. (1994). Evolving 3D Morphology and Behavior by Competition. *Artificial Life*, 1(4):353–372.
- Soros, L. (2018). *Necessary Conditions for Open-Ended Evolution*. PhD thesis, University of Central Florida.
- Spector, L. and Robinson, A. (2002). Genetic Programming and Autoconstructive Evolution with the Push Programming Language. *Genetic Programming and Evolvable Machines*, 3(1):7–40.
- Stanley, K. O. (2010). To achieve our highest goals, we must be willing to abandon them. *ACM SIGPLAN Notices*, 45(10):3–3.
- Stanley, K. O. (2019). Why Open-Endedness Matters. *Artificial Life*, 25(3):232–235.
- Stanley, K. O. and Lehman, J. (2015). *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer, Cham.
- Stanley, K. O., Lehman, J., and Soros, L. (2017). Open-endedness: The last grand challenge you’ve never heard of. *O’Reilly Ideas*.
- Stepney, S. and Hickinbotham, S. (2020). Innovation, variation, and emergence in an automata chemistry. In *ALife 2020, Montreal, Canada (virtual), July 2020*, pages 753–760. MIT Press.
- Szathmáry, E. (2015). Toward major evolutionary transitions theory 2.0. *Proceedings of the National Academy of Sciences*, 112(33):10104–10111.
- Szathmáry, E. and Smith, J. M. (1995). The major evolutionary transitions. *Nature*, 374(6519):227–232.
- Taylor, T. (2019). Evolutionary Innovations and Where to Find Them: Routes to Open-Ended Evolution in Natural and Artificial Systems. *Artificial Life*, 25(2):207–224.
- Taylor, T., Auerbach, J. E., Bongard, J., Clune, J., Hickinbotham, S., Ofria, C., Oka, M., Risi, S., Stanley, K. O., and Yosinski, J. (2016). WebAL Comes of Age: A Review of the First 21 Years of Artificial Life on the Web. *Artificial Life*, 22(3):364–407.
- Thurner, S. (2011). A Simple General Model of Evolutionary Dynamics. In Meyer-Ortmanns, H. and Thurner, S., editors, *Principles of Evolution: From the Planck Epoch to Complex Multicellular Life*, The Frontiers Collection, pages 119–144. Springer, Berlin Heidelberg.
- Thurner, S. (2018). The Creative Destruction Of Evolution. In Sim, S. and Seet, B., editors, *Sydney Brenner’s 10-on-10: The Chronicles of Evolution*. Wildtype Books.
- Thurner, S. (2020). *Die Zerbrechlichkeit der Welt*. edition a, Wien.
- Thurner, S., Hanel, R., and Klimek, R. (2018). *Introduction to the Theory of Complex Systems*. Oxford University Press, New York.
- Thurner, S., Klimek, P., and Hanel, R. (2010). Schumpeterian economic dynamics as a quantifiable model of evolution. *New Journal of Physics*, 12(7):075029.
- Wagner, A. (2011a). The low cost of recombination in creating novel phenotypes. *BioEssays*, 33(8):636–646.
- Wagner, A. (2011b). *The Origins of Evolutionary Innovations*. Oxford Oxford University Press.
- Wagner, C. E., Harmon, L. J., and Seehausen, O. (2012). Ecological opportunity and sexual selection together predict adaptive radiation. *Nature*, 487(7407):366–369.
- Woolley, B. G. and Stanley, K. O. (2014). A novel human-computer collaboration: combining novelty search with interactive evolution. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 233–240.