An Ecological Development Abstraction for Artificial Intelligence

Samuel H. Kenyon
Cambridge, MA 02139
sam@synapticnulship.com

Abstract

A biologically inspired AI abstraction based on phylogenetic and ontogenetic development is proposed. The key factor in the abstraction is iterative development without breaking existing functionality. This approach involves design and observation of systems comprised of artificial organisms situated in environments. An ontogenetic framework involving development and learning stages is discussed. There are critical relationships between iterative biological development and iterative software development. Designing artificial organisms will require a balance of architecture design and design of systems that automatically design other systems. New taxonomies might enable comparisons and sharing of artificial organism design and development.

Natural intelligence entangles mental life, behavior, and environment. Besides the research desire to achieve biological kinds of intelligence in artificial systems, a complimentary worthwhile goal is to expand the domains of AI to include more real world environments where humans and other animals normally operate. So, one might ask what are the causes of intelligence (and lapses thereof) in natural contexts?

A particular animal's behavior at any given time can be explained at several simultaneous levels, which come in three kinds as categorized by (Konner 2010):

1. Remote or evolutionary causation, which includes the genome which in turn is the result of phylogenetic constraints and ecological causes.

2. Developmental causation, which includes the embryonic period where the genome kickstarts ontogeny, formative early-environment effects, and non-formative lifetime environmental effects.

3. Proximate or functional causation, which includes hormonal and metabolic effects, short-term physiology, and elicitors. Short-term physiology, which refers primarily to neural circuits, is the immediate internal cause of behavior. Elicitors, or releasers, are stimuli which cause external behavior.

In this paper I propose to use abstractions of those causes as part of an ecological development approach. Development can be thought of as design in two dimensions: phylogeny and ontogeny. Phylogeny is evolutionary history, indicated by the vertical axis in Figure 1. An ontogeny occupies the lifespan of an organism; multiple ontogenies are represented in Figure 1 as horizontal axes. A special phase in ontogeny is embryogeny, where the genome instantiates the organism. In reference to the aforementioned three kinds of causation, the remote/evolutionary kind of causation happens in phylogeny. Developmental causation and functional/proximate causation happen during ontogeny and embryogeny.

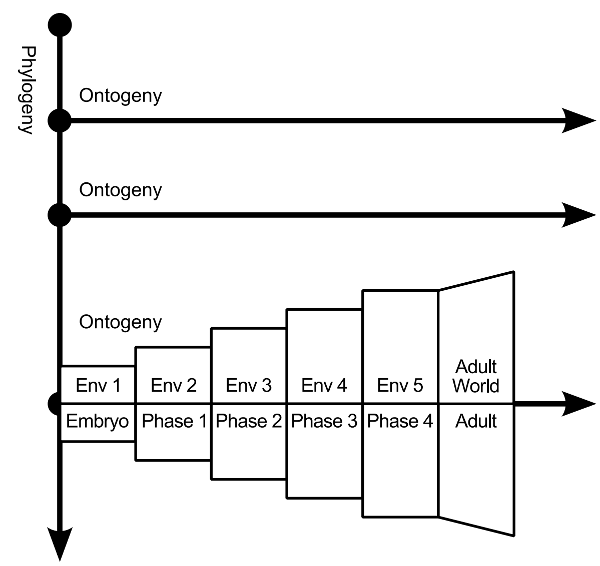

Figure 1: Phylogenetic and ontogenetic dimensions of development

Biologically inspired development as a representation for a system has the potential benefits of compactness, scalability, self-organization, robustness in balance with adaptability, evolvability, fault tolerance, and self-repair (Floreano and Mattiussi 2008). Whereas Floreano and Mattiussi state that development abstractions must include cellular mechanisms, in this paper I propose to abstract away those details.

Those aspects of biological development that are purely in support of the substrate will be abstracted out, such as myelination. However, it is quite possible for artificial substrate constraints to arise. But those should not be so great to make the endeavor impossible. It is certainly possible to evolve and grow software (I will discuss software development later in this paper). The embryonic transformation of information from phylogenetic space to ontogenetic space is actually easier in software than in biology since constraints can be entirely arbitrary in software. Even when using physical robots for embodiment, the entire process can be informational, e.g. autogenerating various electromechanical designs which are input to various machines in the world of the researcher to manufacture the robots which are injected back into the robot's subset of the real world.

The essence of cellular mechanisms that remains in this abstraction is the modification of information systems without breaking existing functionality.

The development abstraction enables valuable AI research in two ways. First, the aforementioned essence can be used as a requirement—for both phylogeny and ontogeny—to build intelligence as extensions to already working intelligence. So it is a way to build potentially complex AI with iterative simple changes.

Second, the abstraction recognizes the holistic nature of human-level intelligence. The intention is to inherently support ecological (Horton, Chakraborty, and St. Amant 2012; Montebelli, Lowe, and Ziemke 2011) and embodied (Anderson 2003) approaches, as well as overlapping enactive approaches, without necessarily excluding extended mind (Clark and Chalmers 1998) ideas that contrast with enactive ideas (Gallagher and Miyahara 2012).

The enactive approach views perceptual experience as an activity of environmental exploration using sensorimotor dependencies and knowledge (Noë 2004). This approach would be difficult to experiment with using most AI abstractions and systems out of the box; only certain types of ALife and mobile autonomous robotics have even the minimal requirement of situated autonomous creatures. Development in an abstraction layer involving issues of an organism growing and living in an ecology provides the minimal configuration to observe and/or purposely design enactive perception and other forms of ecological cognition.

Agents in this development abstraction should typically be autopoietic systems, in which the organism is a phenomenon in itself as opposed to merely a role in population statistics, a notion which historically helped form the enactive approach (Froese 2011b). An artificial organism developed as an autopoietic system ensures that the concept of organism is not just a label for a sum of parts (Froese 2011a). Autopoietic artificial organisms are still developed and interdependent on their ecology, but they are also self-maintaining unified organizations.

Ontogenetic Frameworks

Ontogenetic space contains many kinds of things, such as the artificial organism (which may require an embryonic transformation mechanism to emerge from a digital space into a mixed digital-analog space), environmental agents, and environmental artifacts (including amorphous things like "a mountain"). To work in this abstraction, one must have an ontogenetic framework which is composed of and attempts to keep track of these things (be they real or virtual).

There also must be meta-mechanisms for recording all aspects of the artificial organism's development and interaction with the environment. The meta-mechanisms should store as much state information as possible at high frequency as well as additional substrate logging. Importantly, this information includes internal mental state and the actual mental architecture at a given sample time. Furthermore, these internal states can be correlated with external states. Meta-mechanisms can also be engineered for the ability to inject changes into the framework using new data or already recorded data. The ability to test a single variable change would be quite useful for a complex dynamical system.

Another aspect of ontogeny is the process of embryogeny, in which a genotype is transmogrified into a phenotype. A researcher could choose to use development during ontogeny but skip embryogeny by starting with a neonate. And the starting mental seed does not necessarily have to be evolved; it could come from any method. However, there may be benefits to using the two dimensions (phylogeny and ontogeny) fully in this abstraction. In one experiment, evolution with implicit (more like biology) artificial embryogeny outperformed not only other kinds of artificial embryogeny but also evolution with no embryogeny (Bentley and Kumar 1999).

Development and Learning Stages

Abstract development allows us to say what elements are plastic and at what time they are at a particular plasticity. In this light, learning and development are differentiated by the types of plasticity they use. An artificial organism learns with a plasticity that is more about content and new concepts built on older ones, whereas its development is more about building those underlying concepts and the very machinery needed to be able to learn.

Another ontogenetic framework tool that is particularly relevant is the inclusion of software/hardware created to be a causal contributor to an artificial organism's development. A sequence of development phases could be established—such as designed microcosms of the adult ecological niche—to shape the plastic artificial mind. E.g., the first phase provides a very forgiving environment, the second one is a bit harder and has more challenging toys, and so on until the organism reaches adulthood. Or, for learning, the phases could be established like the increasingly more difficult levels of a video game, presumably increasing the skill of the player at each level.

In Figure 1, the third ontogeny is expanded to indicate a four stage development pattern preceded by an embryogeny stage, culminating in an adult stage ("Env" means environment). The adult stage may include continual development and learning. In each stage, the environment has grown and the artificial organism has grown, including its internal cognitive architecture. Depending on the phenotype, its morphology may have changed at each stage, and if it did, that also affected its cognitive architecture.

I do not specify here what the design space is for an artificial organism to be able to develop and learn in increasingly more challenging stages. However, a good start is plasticity combined with curiosity combined with an environment of appropriate possible affordances. Autonomous development stages have been shown to emerge from a curious learning robot (Oudeyer et al. 2005).

Special Agents

The ecological development abstraction inherently supports agent interactions for development and/or learning. For instance, research into social intelligence could experiment with introductions of specifically designed interactive agents or homogeneous tribes of artificial organisms. Given that social complexity co-evolves with neocortex size in primates (and possibly other mammals) (Dunbar and Shultz 2007), social experiments within ecologies could provide insight on mental structures and scaling for primate-level intelligence

Another special agent is the "caregiver," typically providing maternal interactions in primates. It has been theorized that caregiver interactions are a primary causation of the human capacity for symbolic thought (Greenspan and Shanker 2004). The artificial analogue would serve the purpose of triggering certain mental structures or competences such as emotionally-charged mental symbols for human-level conceptual cognition.

Agents, including but not limited to the aforementioned caregiver, could assist in training. It will become a matter of what static and dynamic elements of the environment to introduce at what times to achieve what developmental effects.

Iterative Evo-Devo

The four "secrets" of evolutionary innovation are (Carroll 2005):

1. Tinkering (work with what is already present).
2. Multifunctionality.
3. Redundancy.
4. Modularity.

These essences should still be valid in a development abstraction, and they apply to both evolution and life history. In this paper I am focusing on number one (tinkering).

The practices of software engineering are relevant, and not just because AI is usually software. In software engineering the commonly used term "development" does not necessarily have anything to do with biological development. Yet, there are some similarities, namely with iterative methodologies.

Many Agile software engineering projects use iterative development in which user stories or tasks are completed in small (e.g. two week) time boxes, which are called sprints in the Scrum variant of Agile. At some point, enough sprints have occurred to make the software product good enough for a release, after which development may or may not continue. The Agile principles state that working software is the primary measure of success and that this working software should be delivered frequently (Beck et al. 2001). One way to realize this is to make the software work at the end of every sprint, even if it is a very immature prototype. Although iterative is often associated with incremental, a purely incremental approach would not necessarily result in a working system after each increment; of course increments may be finer grained than the iterations. Besides being less brittle to change, iterative development ideally eliminates the need for later integration as all changes are also inherently integrations.

Nature uses iterative methods (aka tinkering) in phylogeny and ontogeny. Although ontogeny does not recapitulate phylogeny, there is a shared requirement: the changes of development (and trial-and-error learning) are always potential risks. If the change leaves the organism in a state where it's not viable in its ecological/social context, it could die. Analogically, phylogeny risks fatal nonfunctional phenotypes when it mutates genes (Konner 2010).

Biological phylogenetic change is comparable to software development of artificial organism designs. Biological ontogenetic change is a similar phenomenon to the runtime changes of an artificial organism. Neither of these comparisons should be controversial; but I am saying that beyond those likenesses lie the ability to use the exact same methodology—accretion, which is a kind of iterative development.

In practice, software projects using iterative development methodologies can have varying requirements and mixtures of other methodologies. Here, a hard requirement of functional continuity at all moments in time regardless of maturity must be used.

The accretion is not merely addition, however, as it can modify older structures. The modifications should generally not remove any necessary functionality. Of course, "necessary" depends on the context—what ecological niche will the organism occupy? If there is a shift in context, then it may be plausible to also have a shift in functionalities. Removal in general can be a good method for refinement, e.g. biological removal of overly-connected neural pathways to leave the useful networks. Development of computer programs by humans have similar patterns where a program reaches a certain amount of bloat at which point refactoring, which may involve simplification and reduction, may be necessary to enable further development while maintaining the usefulness of the program.

Modification of ontogenetic structures can result in homologies; i.e. those structures which developed from the same sources but are slightly different in form or function. Homologies would be apparent when comparing an artificial organism to itself in the past (just as one can diff a revision of source code to a previous revision) or to an evolutionary relative. Homologies aren't the only way to modify mind development; new structures have to be created as well.

There are some evolutionary biological tricks that could be used in artificial evolution of development as well, such as facultative adaptations (options that can be switched on/off by environmental stimuli) and heterochrony (timing of developments).

Another dynamic, which heterochrony can affect, is the altricial-precocial spectrum (Starck and Ricklefs 1998). An example of a very precocial animal is a bird that hatches covered with down, eyes open, and can soon leave the nest. In contradistinction, the very altricial version would hatch with no down, eyes closed, and stay in the nest for a long time, dependent on its parents for food. An artificial organism could be made up of a mixture of both types—a particular mental structure (and behavior) could be precocial while another one is altricial. At the level of the entire organism in general, it could have a precocial embryogeny followed by altricial childhood development. The ecologies will determine what structures and skills need to be linked together on the altricial-precocial spectrum. Precocial mental mechanisms, such as bootstrapping learning, could be used in combination with—and be the cause of—altricial skills which learn reusable information chunks (Sloman and Chappell 2005).

Design of Design

For a development abstraction to make progress, it should not be reduced into just a narrow algorithm, such as evolving weights for artificial neural nets. The development abstraction uses life cycles in which autonomous mental architectures grow and learn. The mind is a system, but it is also an architecture, which at many levels of abstraction is composed of heterogeneous modules. To make progress, AI experimenters have to design ecological evo-devo frameworks which design minds. The heterogeneous modules and interfaces that tie them together should be a main focus of concern. And it all should be in the context of iterative development of genotypes and phenotypes. This is a shift from mindless approaches of machine learning and evolutionary ALife.

It was stated over a decade ago that behavioral robots and ALife had reached an impasse, and artificial creatures had never taken off like natural systems (Brooks 2002). This is not the mystery it may seem—it is because of a lack of meta-design in a development abstraction and a lack of architectural thinking. But, one might argue, the whole point of most AI systems inspired by biology is to have the computer figure out the design for you, for instance by evolving the best fit-for-walking robots in micro-worlds. So how does one reconcile this potential design paradox? I suspect the answer is already known by many Agile software developers: a design lives with the thing that is designed. Design is just a derivative view of an abstraction of something. And a document referred to as "the design" is not really the design; at best it's a representation of a static concept of the design which may not even be feasible. There is a balance (and a cycle) between upfront analysis/design versus coding implementations and testing them in the wild. And there is a balance between coarseness of design at a given time—how detailed is it for a particular context? A genotype must balance what details it codes for versus coding for structures that will figure out the details during ontogeny. Complex robust architectures that fit into their ecology emerge this way.

Taxonomies

A taxonomy of artificial embryogeny systems has been proposed before (Stanley and Miikkulainen 2003) as well as taxonomies for robot skills (Huckaby and Christensen 2012). It is conceivable that researchers could collectively contribute to a comparative taxonomy of evo-devo intelligence patterns that can be remixed into new working creatures and/or improved disembodied computer systems.

Comparisons and benchmarks can be difficult for AI, and it makes no sense to evaluate organisms from disparate niches with the same standardized test. Intelligence in one context does not necessarily result in intelligence in others. Perhaps instead biological-style taxonomies and comparisons could be of use to AI practitioners using a developmental abstraction. Although the practitioners are also the designers at some meta-level, certainly they will observe unexpected emergent phenomena which could be compared and classified.

Taxonomies could define toolboxes of genotype modules which result in known good ontogenetic developments which in turn create known good mental structures or behaviors in particular ecological contexts. These modules could in turn can be used in new designs via composition. Of course, modular AI doesn't require an ecological developmental approach. But, for example in (Duro, Becerra, and Santos 2000), the nature of idiosyncrasies between robots limits the re-usable modules to features of the least capable robot. And as with all modular systems, nontrivial reuse requires flexible interfaces. These two observations indicate that an adaptive developmental system might actually be more suited towards reuse of informational modules (although the "modules" may be genetic-like instructions on how to construct modules). As homologies, the reusable modules may have radically different forms depending on the artificial organism that uses it.

Conclusion

I have attempted to synthesize an AI abstraction based on phylogenetic and ontogenetic development. Many biological details such as cellular mechanisms are abstracted out. This approach involves design and observation of ecologies containing evolved artificial organisms. The intelligence of an artificial organism is not just inside its brain analogue, but also a property of an organism-environment system.

A useful ontogenetic framework might involve a series of increasingly more advanced environments (and possibly social interactions) to go along with an artificial organism's child-like development and learning stages. Some of these stages may also include special agents, such as a caregiver to provide interactions needed for mental growth.

I also showed relevant relationships between iterative biological development and iterative software development. Designing artificial organisms will require a balance of architecture design and design of systems that automatically design other systems. Taxonomies could be made to enable comparisons of artificial evo-devo patterns amongst AI researchers.

Working with systems does not prevent focused research once the system is running. As a very speculative example, if vision is primarily an aspect of the sensorimotor behavior system, then the developmental approach enables an inherent methodology to always involve the sensorimotor-environment loop and avoid non-affordances. If a perception is not related to a bodily-mental use, then the organism will not develop that perception.

Enactive (Noë 2009), embodied (Vernon 2008), and "more" Heideggerian (Dreyfus 2007) approaches to intelligence and consciousness lead to the possibility that there cannot be humanlike cognition aside from an organism dynamically existing—in and as part of—its environmental niche. Therefore, an ecological development abstraction could be key to making humanlike artificial intelligence. This paper is merely a proposal, however, and one AI tradition that should go on if all others die is the desire to implement. Therefore, much meta-development and development remains to see if a development abstraction is the right way to produce artificial creatures which live as natural creatures do, and if this is the right way to view intelligence.

References

Anderson, M. L. 2003. Embodied cognition: A field guide. *Artificial Intelligence* 149(1):91 – 130.

Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R. C.; Mellor, S.; Schwaber, K.; Sutherland, J.; and Thomas, D. 2001. Manifesto for agile software development. http://agilemanifesto.org.

Bentley, P., and Kumar, S. 1999. Three ways to grow designs: A comparison of evolved embryogenies for a design problem. In *Genetic and Evolutionary Computation Conference*, 35–43. Morgan Kaufmann.

Brooks, R. A. 2002. *Flesh and Machines: How Robots Will Change Us*. New York: Pantheon Books.

Carroll, S. B. 2005. *Endless Forms Most Beautiful: The New Science of Evo Devo*. New York: Norton.

Clark, A., and Chalmers, D. 1998. The extended mind. *Analysis* 58(1):10–23.

Dreyfus, H. L. 2007. Why heideggerian ai failed and how fixing it would require making it more heideggerian. *Philosophical Psychology* 20(2):247–268.

Dunbar, R. I. M., and Shultz, S. 2007. Understanding primate brain evolution. *Philosophical Transactions of the Royal Society B: Biological Sciences* 362(1480):649–658.

Duro, R. J.; Becerra, J. A.; and Santos, J. 2000. Improving reusability of behavior based robot cognitive architectures obtained through evolution. In *Proceedings of the 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, 35–43. Noordwijk, The Netherlands: ESA.

Floreano, D., and Mattiussi, C. 2008. *Bio-inspired Artificial Intelligence: Theories, Methods, and Technologies*. Cambridge, Mass.: MIT Press.

Froese, T. 2011a. Breathing new life into cognitive science. *Avant. The Journal of the Philosophical-Interdisciplinary Vanguard* 2(1):113–129.

Froese, T. 2011b. From second-order cybernetics to enactive cognitive science: Varela's turn from epistemology to phenomenology. *Systems Research and Behavioral Science* 28(6):631–645.

Gallagher, S., and Miyahara, K. 2012. Neo-pragmatism and enactive intentionality. In Schulkin, J., ed., *Action, Perception and the Brain: Adaptation and Cephalic Expression*. New York: Palgrave Macmillan.

Greenspan, S. I., and Shanker, S. 2004. *The First Idea: How Symbols, Language, and Intelligence Evolved From Our Early Primate Ancestors to Modern Humans*. Cambridge, MA: Da Capo Press.

Horton, T. E.; Chakraborty, A.; and St. Amant, R. 2012. Affordances for robots: a brief survey. *Avant. The Journal of the Philosophical-Interdisciplinary Vanguard* 3(2):70–84.

Huckaby, J., and Christensen, H. 2012. A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics. In *AAAI Workshops*.

Konner, M. 2010. *The Evolution of Childhood: Relationships, Emotion, Mind*. Cambridge, Mass.: Belknap Press of Harvard University Press.

Montebelli, A.; Lowe, R.; and Ziemke, T. 2011. Energy constraints and behavioral complexity: The case of a robot with a living core. In *AAAI Fall Symposium Series*.

Noë, A. 2004. *Action in Perception*. Cambridge, Mass.: MIT Press.

Noë, A. 2009. *Out of Our Heads: Why You Are Not Your Brain, and Other Lessons from the Biology of Consciousness*. New York: Hill and Wang.

Oudeyer, P.-Y.; Kaplan, F.; Hafner, V. V.; and Whyte, A. 2005. The playground experiment: Task-independent development of a curious robot. In *Proceedings of the AAAI Spring Symposium on Developmental Robotics*, 42–47.

Sloman, A., and Chappell, J. 2005. The altricial-precocial spectrum for robots. In *Proceedings IJCAI '05*, 1187–1192.

Stanley, K. O., and Miikkulainen, R. 2003. A taxonomy for artificial embryogeny. *Artif. Life* 9(2):93–130.

Starck, J. M., and Ricklefs, R. E. 1998. Patterns of development: The altricial-precocial spectrum. In Starck, J. M., and Ricklefs, R. E., eds., *Avian Growth and Development: Evolution Within the Altricial-Precocial Spectrum*. New York, NY: Oxford University Press. 3–30.

Vernon, D. 2008. Cognitive vision: The case for embodied perception. *Image Vision Comput.* 26(1):127–140.