Evolvability of Personal Learning Environments

Chrystopher L. Nehaniv

Professor of Mathematical & Evolutionary Computer Sciences
School of Computer Science
Faculty of Science, Technology & Creative Arts
University of Hertfordshire,
Hatfield AL10 9AB, UK
+44 1707 284470
C.L.Nehaniv@herts.ac.uk

ABSTRACT

We review the concepts of Darwinian evolution and evolvability, and discuss the extent to which these can be brought to bear on the problems of personal learning environments (PLEs). While it is problematic to identify an evolving population of *individuals* (a definitional requisite of Darwinian evolution) in artifacts, we suggest an instance of a PLE system as fielded can play the role of individual in this seting, while configuration, code and component organization can play the role of *inheritable genetic information*. Also discussed are adaptivitiy, plasticity, robustness, and evolvability in this setting, as well as the role of *sex* (transfer of inheritable information from one individual to another) in providing plasticity in a community of use in the context of changing requirements.

Keywords

Evolvability, Sex, Personal learning environment, Plasticity, Darwinian evolution.

1. INTRODUCTION: DARWINAN EVOLUTION AND EVOLVABILITY FOR ARTIFACTS?

To what extent does it make sense to apply the biological notion of evolution to artifacts like software systems? Darwinian evolution is a process undergone by a changing population of individuals, in a 'struggle for existence' in which better adapted variants are more likely to survive, reproduce, and have their character traits persist beyond the lifespan of single individuals. The Darwin-Wallace theory of evolution has revolutionized Biology. It has even given philosophers a way to explain 'purpose' and 'meaning' within a mechanistic framework. And it has given rise to effective methods to apply these ideas to the automatic design of artificial systems, ranging from engineering optimization, to aeronautic, architectural and even artistic design by instantiations of principles (or axioms) that capture essential aspects of such dynamical processes, and has even been applied to the design of pharmaceuticals and of molecules with particular enzyme-like properties. Entire areas of computer science such as genetic algorithms and evolutionary computation have grown up to exploit this; and the evolvability of Darwinian processes and how it can be supported is a vibrant area of inquiry for both natural and artificial evolutionary processes.

This impressive success of Darwinian theory in this areas suggests that there might similar dynamics in other areas such as the 'evolution' of ideas, tools, artifacts, culture, or software systems. But does it actually make sense to apply Darwinian terms such as 'fitness' and 'evolvability' in these settings? To what extent are the axioms of Darwinian theory valid or even interpretable in these domains? Does the theory have any explanatory or predictive power and can it guide us in our design of artifacts and software systems? What about interactive systems that humans use in contexts of changing requirements? – here human beings play special roles in the mechanisms of inheritance, variability and in any notion of fitness.

We have been exploring the problems and issues that arise with attempting to apply the theory of evolution to realms outside biology. Key issues and pitfalls preventing the direct application of Darwinian dynamics to other domains are the identification of individuals (members of a population on which the dynamics operates) and the inheritability of fitness by offspring. Despite these difficulties and the divergences with biological evolution, and further research needed into evolvability in all domains, one can identify an entire array of important parallels and concepts from biological evolution which are or can be used to inform the design of adaptive, interactive artifacts and software systems.

2. SCALES OF PERSISTENCE AND HEREDITY

Darwinian evolution systems are comprised by populations of individuals undergoing processes of inheritance (in producing offspring) , variation and selection. If individuals cannot be clearly identified then application of this theory is not likely to be conducted rigorously. However, weaker analogues of evolution occur on a spectrum in which there is any sort of *descent with modification* on the one end, but evolving populations of individuals at the other along a scale toward full-blown Darwinian evolution:

- persistence without change, growth, or variation - e.g. of a stone existing without substantial change over a long period of geological time;

- growth and spread without variability, e.g. in the growth of crystals;
- persistence with growth and variation *(lifespan of single living things, maintained software and robotic systems, coral reefs, cities, and many other entities):* persistence and variability providing analogues of heritability but not actual reproduction;
- examples closer to biological evolution acting on populations but still lack well-defined self-reproducing individuals. e.g. design and cultural traditions, and generations of software releases;
- Darwinian evolution: heritable variability and fitness in populations of reproducing individual entities. E.g . organic biological evolution of life on earth.

3. FITNESS & EVOLVABILITY FOR SOFTWARE

Fitness and evolvability of software have multiple components which include:

- *Functional properties* (adaptedness to requirements and context of use)
- *Non-functional properties*
- *Variational / Lineage Properties* – capacity to vary / be varied robustly and adaptively
  [NB: The latter properties doe not effect the immediate fitness, but crucial to evolvability!]

4. SOFTWARE EVOLUTION, REQUIREMENTS CHANGE & EVOLVABILITY

In software engineering, change in requirements and context of use is the major factor in cost and impacts the areas of requirements engineering, software maintenance, and software evolution.

Evolvabilty for artifacts is the capacity of the systems, organizations and networks producing them to give rise to adaptive variants that flexibly meet changing requirements over the course of long-term change (Nehaniv et al. 2006).

Evolvability as a capacity to generate adaptive variability in tandem with continued persistence of software artifacts would be welcome in software engineering.

5. PLES AND APPLICATIONS TO SOFTWARE MAINTENANCE

As software maintenance costs exceeds 80% of all software costs, even small advances via the application of Darwinian theory to software could well result in the savings of billions or trillions of euros annually. One avenue worth exploring is the application of PLEs to support communities programmers and stakeholders in the creation and deployment of software, during the course software evolution and (inevitable) requirements change as contexts of use change. Conversely, PLEs themselves can be members of evolving populations whose evolvability and plasticity properties deserve the attention of those who build them or advocate their use in various settings.

6. SOFTWARE EVOLUTION ANALOGUES TO BIOLOGICAL EVOLUTION

Features identified in software evolution that may enhance evolvabilty (including maintainability and adaptivity to requirements changes) are the following:

- Re-use (not replication)
- Modularity (Parnas)
- Information Hiding
- Encapsulation
- Object-oriented “inheritance”
- Appropriate coupling and cohesion (Dijkstra, Parnas)
- Abstract Data Types (Goguen)
- Engineering for robustness to requirements change (e.g. Goguen, Berners-Lee)
- Dynamically configurable collections of interacting components (analogous to cellular organization) in differentiated multicellular dorganisms)

7. PERSONAL LEARNING ENVIRONMENTS (PLE) EVOLUTION & EVOLVABILITY IN A DARWINIAN FRAMEWORK

A suggestion for how to bring Personal Learning Environments into the Evolutionary Frame is to consider:

- PLE ‘system as fielded’ (instance) could be considered an *individual.*
- A system as fielded persists through time, although it may change, into a new fielded system due to adding or removing components, etc., this results in descent with modification which can be viewed either as a case of (*vertical) heredity* or as the *development* of an individual over time.
- *Inheritance:* its lines of code or, better, its constituent modules might be considered as ‘genes’ (potentially inheritable – re-useable – in other PLEs, and could be copied or imitated by new fielded instances of PLEs).
- Variation: (1) customization of a generic software product via parameters and installation, components options / apps; (2) copying / sharing from others’ PLE settings. Change in context of use and thus changing interactions and requirements will provide *phenotypic variation* that must be supported by *phenotypic plasticity*, the capacity to adapt and change robustly.
- Iteratively adapted by users to learning context & changing requirements -> evolution
- Capability to generate adapted organizational instances adapted to the current user requirements: *evolvabilty of PLEs*

8. SEX: THE TRANSFER OF INHERITABLE MATERAIL

Sex in the biological sense in evolutionary theory, is the exchange or transfer of heritable genetic material from one individual to another one. It is well known in evolutionary studies for its potential to increase the rate of evolution by creating variation.

Copying configuration or component organization from one or more PLEs to an existing PLE system, or creating a new with this inheritable information from several PLEs constitute natural and potential very useful examples of sex.

Note that genetic variability via sex in PLEs will in large part be driven by human preferences and choices, as well as fashions and trends. Communities of practice will share basic skeletal configurations and costumize these by learning what components to bring in from other members of social communities that share practices. Sharing such configuration information and component organization is thus a pervasive form of sex in PLE evolution.

9. ACKNOWLEDGMENTS

The author thanks Dr. Effie Law and the other organizers of the First Workshop on Exploring the Fitness & Evolvability of Personal Learning Environments (EFEPLE11)30-31 March 2011, at the 2nd STELLAR Rendez-Vous, French Alps for the opportunity to present this work in an invited keynote talk.

10. REFERENCES

[1] Nehaniv, C. (2003). Evolvability, *Biosystems: Journal of Biological and Information Processing Systems*, 69(2-3), 77-81.

[2] Nehaniv, C., Hewitt, H., Christianson, B., & Wernick, P. (2006). What software evolution and biological evolution don't have in common. In *Proc. of 2nd Int'l IEEE Workshop on Software Evolvability (SE'06).*

[3] Darwin, C. R. and A. R. Wallace. 1858. On the tendency of species to form varieties;
and on the perpetuation of varieties and species by natural means of selection.
[Read 1 July] *Journal of the Proceedings of the Linnean Society of London.. Zoology* 3: 46-50.

[4] Dijkstra, E. W. 1968, Goto statement considered harmful, *Communications of the Association for Computing Machinery*, 11(3):147-148.

[5] Parnas, D. 1972, On the criteria to be used in decomposing systems into modules, *Communications of the Association for Computing Machinery*, 15(2):1052-1058.

[6] Wagner, G. P. & Altenberg, L. 1996, Complex Adaptations and the Evolution of Evolvability, *Evolution*, Vol. 50, No. 3, pp. 967-976, June.

[7] Nehaniv, C.L. (2000). Evolvability in Biological, Artifacts, and Software Systems. In: C. C. Maley and E. Boudreau (Eds.), *Artificial Life 7 Workshop Proceedings - Seventh International Conference on the Simulation and Synthesis of Living Systems*, Reed College, pp. 17-21.

[8] Nehaniv, C. L. (2005). Self-Replication, Evolvability and Asynchronicity in Stochastic Worlds C. L. Nehaniv, "Self-Replication, Evolvability, and Asynchronicity in Stochastic Worlds", *Proc. 3rd Symposium on Stochastic Algorithms, Foundations and Applications (Symposium Stochastische Algorithmen, Grundlagen und Anwendungen) - SAGA'05, Moscow, Russia, October 20-22, 2005*, Editors: Oleg B. Lupanov, Oktay M. Kasim-Zade, Alexander V. Chaskin, Kathleen Steinhöfel, Lecture Notes in Computer Science, vol. 3777, Springer Verlag, 2005, pp. 126-169. Corrected version on-line at URL: http://homepages.feis.herts.ac.uk/~comqcln/nehaniv-SAGA05-withcorrections.pdf

[9] Goguen, J.A., A Categorical Manifesto, *Mathematical Structures in Computer Science*, **1**(1):49–67, 1991.

[10] Conrad, M. 1990, The geometry of evolution. *Biosystems* 24:61-81.

[11] Goguen, J. 1994, Requirements Engineering as the Reconciliation of Technical and Social Issues, in *Requirements Engineering: Social and Technical Issues*, edited Marina Jirotka and Joseph Goguen, Academic Press, 1994, pp. 165-199.

[12] Goguen, J. 1996, Formality and Informality in Requirements Engineering, Proceedings, *Fourth International Conference on Requirements Engineering*, IEEE Computer Society, April 1996, 102-108.

[13] West-Eberhard, M. J. 1998, Evolution in the light of developmental biology, and *vice versa*, *Proc. Natl. Acad. Sci. USA* 95:8417-8419.

[14] Kirschner, M. and Gerhart, J. 1998. Evolvability, *Proc. Natl. Acad. Sci. USA*, 95:8420-8427.

[15] M. M. Lehman (2005), "The role and impact of assumptions in software development, maintenance and evolution", *First International IEEE Workshop on Software Evolvability (SE'05 - Budapest, Hungary)*, pp. 3-14, 2005.

[16] Berners-Lee, T., 1998, "Evolvability", On-line lecture slides from WWW7, 1998. http://www.w3.org/Talks/1998/0415 Evolvability/ [last accessed 1 August 20007]