

Evolution as a Noun and Evolution as a Verb

Submitted to SOCE 2000 Workshop on Software and Organisation Co-evolution, July 13 - 14, 2000,
Imperial College, London

M M Lehman J F Ramil G Kahen

Department of Computing

Imperial College of Science, Technology and Medicine

180 Queen's Gate, London SW7 2BZ, UK

tel + 44 (0) 20 7594 8216; fax + 44 (0) 20 7594 8215

{mml,ramil,gk}@doc.ic.ac.uk <http://www-dse.doc.ic.ac.uk/~mml/feast>

ABSTRACT

The paper starts by distinguishing between two types of software evolution studies. The first addresses evolution as a *noun* and focuses on it as a *phenomenon*. The second addresses the topic as a *verb* and focuses on methods and tools intended to facilitate software evolution and the tasks that implement it. Both views are necessary to achieve increased insight and a degree of mastery of the evolution phenomenon and to develop improved tools and technology.

Keywords: co-evolution, noun and verb, management, software evolution, software process, tools

1 INTRODUCTION

The universal experience that software that is used undergoes continuing evolution was first discussed in the early seventies [leh74]. More recently the theme of evolution has roused widespread interest as demonstrated by various workshops, conferences and research investigations. In the context of this activity it is apparent that those involved divide into two groups. The first and larger group is concerned with the question "*how*", that is, they focus on theories, abstractions, languages, activities, methods and tools required to effectively and reliably evolve a software system. A smaller number of workers are concerned with the question "*what*" and investigate the nature of software evolution and the properties of the phenomenon described by that term. We suggest that the first group consider the term evolution as a *verb*, the second as a *noun*. We further suggest that study of evolution as a verb will benefit from a better understanding of what evolution is and what properties it has, that is, a study of evolution as a noun. In fact, both are required if the community is to make progress in understanding "*why*" and in mastering software evolution.

2 THE WAY FORWARD

Why are both these important? It has long been known that for any system used by more than a handful of users to remain satisfactory for any length of time requires it to be continuously adapted and enhanced. The reasons for this are now well understood. That is, as encapsulated in the first and six laws of software evolution [leh74 to leh00], the phenomenon of evolution is intrinsic to *E*-type systems, that is, systems that are embedded in and actively used in a real world domain [leh85]. Software is playing an increasingly important role in most business and organisations the world over. As computers penetrate ever more deeply into organisational activity the organisation becomes dependent on the satisfactory working of its software. And as the various activities and the domains within which they operate change and evolve the software must be adapted to remain faithful to the changing operational domain and to its needs. It must be continually evolved. Moreover, and as reflected by the present workshop title "software and organisation co-evolution", it is now recognised that the rate at which businesses or other organisations can improve their processes to take full advantage of information technology depends to a significant degree on the rate at which the software can be evolved. The name of the game has become *co-evolution* of business and software processes.

Achievement of organisational effectiveness requires consideration of several evolving entities. The organisation, the market place and other domains in which it operates, the technologies it uses, the management structure that guides and drives it from day to day, the human component executing the many tasks to be performed in pursue of the organisational objectives and the software required to support all this, interact and must not be considered in isolation from one another. The way forward is to harmonise the different domains involved. In order to accomplish such harmonisation a full set of issues and problems need to be systematically addressed. We expect many of these to be reflected by the submissions to this workshop and these will divide into the two categories identified above.

Some workers, for example [kem99,mit00] and [leh74 to 00b], have approached the study of software evolution mainly as a noun. In FEAST and its precursors the approach has been based on observation of industrial processes, metric collection from a number of industrially evolved systems, an approach which follows the more generally scientific method. The analysis and interpretation undertaken in the FEAST studies and in earlier activity revealed the existence of behavioural similarities in the evolutionary attributes of released-based systems. These led to the formulation of *laws* of software evolution, which are attributes of evolution as a noun [fea00]. As shown in a contribution to the FEAST 2000 workshop which precedes the SOCE workshop, these attributes lead to quite specific recommendations for software evolution management [leh00b].

Consider, for example, the use of tools for software evolution. These must be identified and defined in terms of both, a perception of *why* and *how* the process is evolving (nounal) and *how* (the way in which) it is evolved (verbal). In some circumstances, and for some tools, the former view is dominant. For example, in planning evolution studies have suggested [fea00] that, *ceteribus paribus*, the growth per release must be limited if quality and timely delivery problems are to be avoided. The limit depends on past history as well as the nature of the changes. A tool to support management decision in a planning situation will be based on understanding *what* factors influence the desirable growth limit and the consequences of the limit being exceeded. This is primarily based on the noun view of evolution. A tool that would support the derivation of the set of changes to be applied to a system from a formal statement of the changes applied to the specification of that system [tur00] is primarily based on the verb view of evolution. In the long term, both views are necessary and must feed each other to achieve increased insight and a degree of mastery of the phenomenon and to develop improved tools and technology.

ACKNOWLEDGEMENTS

Financial support from UK EPSRC, grant no. GR/M44101 (FEAST/2) is gratefully acknowledged.

REFERENCES¹

- [kem99] Kemmerer CF and Slaughter S, *An Empirical Approach to Studying Software Evolution*, IEEE Trans. on Softw. Eng., vol. 25, n. 4, July/August 1999, pp. 493 - 509
- [leh74]* Lehman MM, *Programs, Cities, Students, Limits to Growth?*, Inaugural Lecture, in Imperial College of Science and Technology Inaugural Lecture Series, vol. 9, 1970, 1974, pp. 211-229. Also in *Programming Methodology*, (D. Gries. ed.), Springer Verlag, 1978, pp. 42-62.
- [leh85] Lehman MM and Belady LA, *Software Evolution - Processes of Software Change*, Academic Press, London 1985
- [leh00] Lehman MM, *Rules and Tools for Software Evolution and Management*, Pre-prints of FEAST 2000 International Workshop on Feedback in Software and Business Processes, July 10 -12, 2000, Imperial College, London, <http://www-dse.doc.ic.ac.uk/~mml/f2000>
- [mit00] Mitleton-Kelly E and Papaefthimiou M-C, *Co-Evolution of Diverse Elements Interacting within a Social Ecosystem*, Pre-prints of FEAST 2000 International Workshop on Feedback in Software and Business Processes, July 10 -12, 2000, Imperial College, London, <http://www-dse.doc.ic.ac.uk/~mml/f2000>
- [fea00] FEAST/2, *Feedback, Evolution and Software Technology*, Project Web Site <<http://www-dse.doc.ic.ac.uk/~mml/feast>>
- [tur00] Turski WM, *An Essay on Software Engineering at the Turn of Century*, working paper, autumn 1999, invited lecture, ETAPS 2000 - European Joint Conf. on Theory and Practice of Software, Technical Univ. of Berlin, March 25 - April 2, 2000

¹ Reprinted in [leh85]