Software Evolution, its Nature and Control - Plans for Further Work after FEAST

WESS 2001
Florence
9 Nov 2001

M M Lehman
Dept. of Computing
Imperial College
180 Queen's Gate
London SW7 2BZ, UK
mml@doc.ic.ac.uk
http://www.doc.ic.ac.uk/~mml

J F Ramil
Dept. of Computing
Faculty of Maths and Computing
The Open University
Milton Keynes MK7 6AA, UK
j.f.ramil@open.ac.uk
http://mcs.open.ac.uk/jfr46

The 'What' & 'Why' of Evolution - FEAST Results

- **Evolution characteristics** and **behavioural patterns** of released-based systems observed in the 90s, similar to those studied in the 70s
- **Support**, after minor modifications, for 7 of the 8 **laws of software evolution**
- New **insights** into and **conclusions** about the process and about its feedback system nature
- Black box and system dynamics **models** to support **process planning**, **management**, **improvement**, **tuning**
- Some 35 **management** and **planning** rules and guidelines

Further investigations being proposed

Two proposed investigations

- With completion of FEAST/1 and /2

EPiCS - Empirical Phenomenology in Component-intensive Software

SETh - Towards a Theory of Software Evolution

Some highlights...

Trend to Component-intensive Software

- A **widespread** view is that the way forward in software development and acquisition is through the **use of components**, as exemplified by COTS
- Not new - discussed at Garmisch conference 68

But unanswered questions in context of evolution phenomenon

Components of E-type Host Systems also Evolve

- ***E*-type** components
  - **solve** a **problem** or address an application in a **real-world domain**
  - **stakeholders' satisfaction**, primary criterion of success
  - as for *E*-type systems, need for **continuing evolution**, **inescapable**

- ***S*-type** components
  - **satisfaction** of an **specification**, primary criterion of success
  - "**bricks**" from which systems are built
  - **acquire** *E*-type characteristics when **integrated**, since it requires assumptions about properties of other components

- Thus, **all** components of *E*-type host systems are subject to evolutionary pressure

- Achieving **evolution by "glueware",** - e.g., component wrapping - diminishes need for component evolution, by **reverting** to **traditional practice**

The Plan

- Assessment of **long-term effectiveness** of component-orientation
- Laws and other FEAST, and related, results as **hypotheses** for empirical investigation
- Study both **quantitative** and **non-quantitative** data
- Investigation will face technical **challenges** - scarcity of data, probably stronger role of individuals in smaller integration teams, increase number of agents, and so on
- Our WESS position paper describes plans for the EPiCS empirical investigation

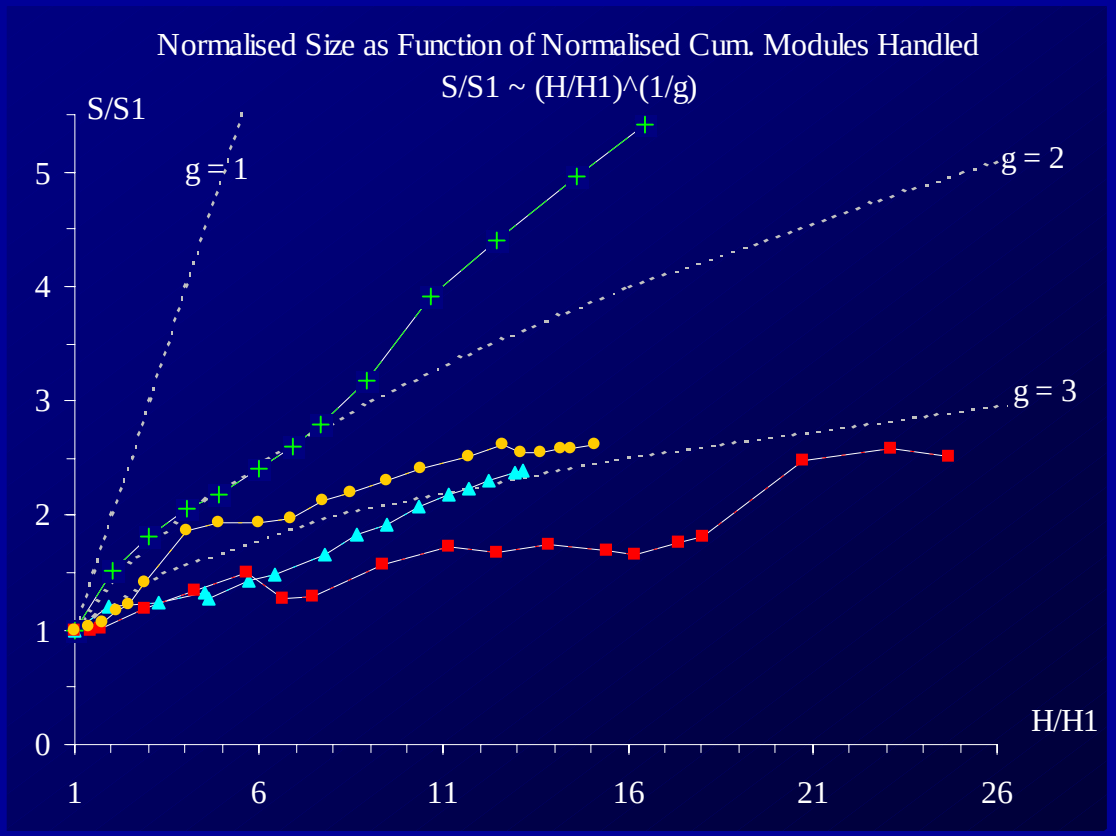
In general, if funded, **EPiCS** project will investigate **system evolution** under component-intensive paradigm

Why a Theory?

- Theory provides **basis** and **framework** for unifying, verifying, extending an engineering technology
- **Need** for a software theory already identified at Garmisch 68
- **Existing** theory addresses **programming methodology**, not industrial **process**
- Software **process improvement** ***ad hoc***, individually conceived, based primarily on **experience**, **intuition**
- Theory seeks, in general, to establish **links** between **descriptions** of facts and **prescription** - when achieved, offers basis for **justification** of ***good practice***

If funded, **SETh** will seek and develop foundations of a theory

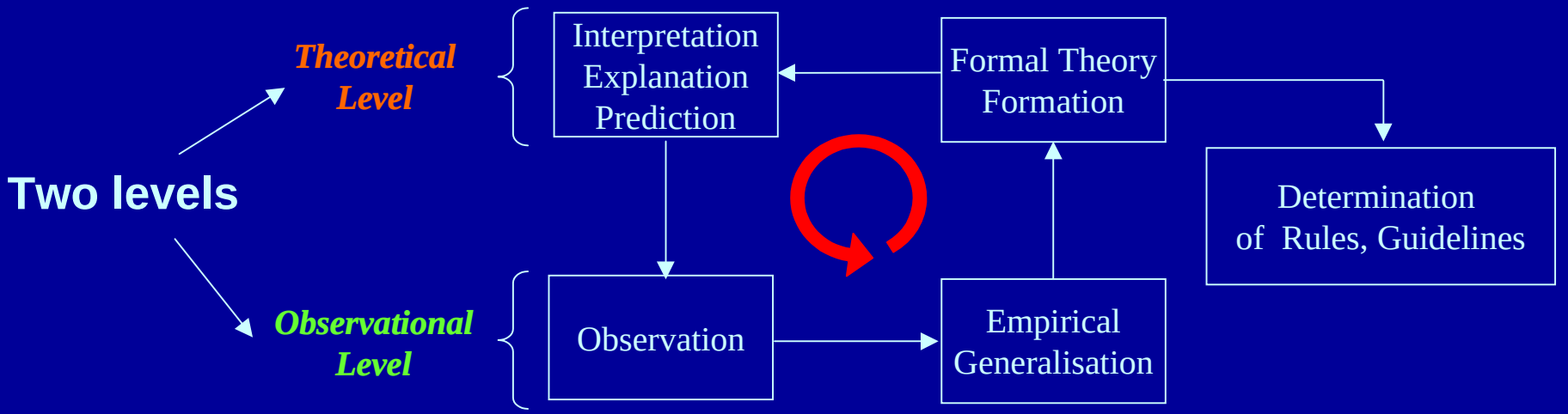
Candidate for Empirical Generalisation



- Growth models discussed in ICSM 2001 paper
- One system starts following $S = E^{1/2}$ trajectory, though mid-life change requires further analysis
- Note closeness in 3 of the systems to "$S = E^{1/3}$ trajectory"

The Plan

- Two-level approach to theory formation



Final Remarks

- FEAST was an attempt to apply **scientific method** to issues usually regarded as not amenable to systematic study
- Its success and the results produced have identified a need for further investigation - hence the two pronged SETh and EPiCS proposals
- In particular, FEAST has provided ***empirical generalisations*** that are essential for development and assessment of candidates theories and for reasoning about the challenges arising from the use of components
- Go-ahead depends on funding decision
- Others groups invited to take up approach
- In particular, we welcome comments from WESS participants with regards to these views and plans

References

- [bar67] BARNEY, G.G. and STRAUSS, A.L.: 'The Discovery of Grounded Theory: Strategies for Qualitative Research', Aldine de Gruyter, 1967
- [bro98] BROWN, A. and WALLNAU, K.: 'The Current State of Component-Based Software Engineering (CBSE),' IEEE Software, September 1998, pp. 37-47
- [car00] CARNEY, D., HISSAM, S.A. and PLAKOSH, D.: 'Complex COTS-based Software Systems: Practical Steps for Their Maintenance', J. of Softw. Maintenance: Res. and Pract., vol. 12, 2000, pp. 357 - 376
- [cho80] CHONG HOK YUEN, C. K. S.: 'Phenomenology of Program Maintenance and Evolution', PhD thesis, Dept. of Comp., Imperial College, 1981
- [cle93] CLEVELAND, W.S.: 'Visualizing Data', Hobart Press, Summit, NJ, 1993, pp. 360
- [dso99] D'SOUZA, D. and WILLS, A.C.: 'Objects, Components, and Frameworks with UML: The Catalysis Approach' (Addison-Wesley, Boston, Ma. 1999)
- [feast] FEAST: 'Feedback, Evolution And Software Technology' Web Site including selected pubs. http://www-dse.doc.ic.ac.uk/~mml/feast <as of Aug. 2001>
- [fen97] FENTON, N.E. and PFLEEGER, S.L.: 'Software Metrics: A Rigorous and Practical Approach', Int. Thomson Comp. Press, London, 1997, pp. 638
- [fesbp] FEAST 2000: 'Intl. Workshop on Feedback and Evolution in Business and Software Processes'. 10 - 12 July 2000, Imp. Col., London. Pre-prints available from http://www.doc.ic.ac.uk/~mml/f2000
- [ffse] FFSE 2001: 'Intl. Special Session on Formal Foundations of Software Evolution', 13 March 2001, Lisbon, http://prog.vub.ac.be/poolresearch/FFSE/FFSE-Workshop.html <as of August 2001>
- [for61] FORRESTER, J.W.: 'Industrial Dynamics' (MIT Press, Cambridge MA, 1961)
- [hal97] HALLSTEINSEN, S. and PACI, M.: 'Software Evolution and Reuse'. Springer Verlag, Berlin, 1997, 293 pp.
- [han96] HAND, D.J. and CROWDER, M: 'Practical Longitudinal Data Analysis', Chapman & Hall, London, 1996, pp. 232
- [hyb97] HYBERTSON, D.W., ANH, D.T. and THOMAS, W.M.: 'Maintenance of COTS-intensive Software Systems,' Softw. Maint.: Res. and Pract., 1997, 9, pp. 203-216
- [iee00] IEE Proceedings - Software: 'Special Issue on Component Based Software Engineering', v. 147, n. 6, Dec. (2000).
- [ispse] ISPSE 2000. 'Proc. of the Intl. Symp. on Principles of Softw. Evolution', Nov. 1-2, 2000, Kanazawa, Japan
- [jab97] JACOBSON, I., GRISS, M. and JONSSON, P.: 'Software Reuse - Architecture, Process and Organisation for Business Success' (Addison-Wesley, 1997, 560 pps.)
- [kah00] KAHEN, G., LEHMAN, M.M., RAMIL, J.F. and WERNICK, P.D.: 'An Approach to System Dynamics Modelling in the Investigation of Policies for E-type Software Evolution.' ProSim 2000, 12-14 July 2000, Imperial College, London UK. A revised version to appear in J. of Syst. and Softw., vol. 15, 2001
- [kem99] KEMERER, C.F. 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
- [kon96] KONTIO, J.A.: 'Case Study in Applying a Systematic Method for COTS Selection'. Proc. ICSE 18, 25-29 March 1996, Berlin, pp. 201-209

References (cont.)

- [leh74] *LEHMAN, M.M.: 'Programs, Cities, Students, Limits to Growth?' Inaugural Lecture, May 1974. Publ. in Imp. Col. of Sc. Tech. Inaug. Lect. Ser., 9, 1970 - 1974, pp. 211-229. Also in Programming Methodology, D Gries (ed.), Springer Verlag, 1978, 42-62
- [leh78] *LEHMAN, M.M.: 'Laws of Program Evolution - Rules and Tools for Program Management'. Proc. Infotech State of the Art Conf., Why Software Projects Fail, - April 1978, 11/1-11/25
- [leh80a] *LEHMAN, M.M.: 'On Understanding Laws, Evolution and Conservation in the Large Program Life Cycle,' J. of Sys. and Software, 1980, 1, (3), pp. 213-221
- [leh80b] *LEHMAN, M.M.: 'Programs, Life Cycles and Laws of Software Evolution'. Proc. IEEE Special Issue on Softw. Eng., 68, (9), Sept. 1980, pp. 1060-1076
- [leh85] LEHMAN, M.M. and BELADY, L.A.: 'Program Evolution, - Processes of Software Change', (Acad. Press, London, 1985)
- [leh89] LEHMAN, M.M.: 'Uncertainty in Computer Application and its Control Through the Engineering of Software,' J. of Softw. Maintenance: Res. and Practice, 1, (1), September 1989, pp. 3-27
- [leh90] LEHMAN, M.M.: 'Uncertainty in Computer Application'. Tech. Let., CACM, 33, (5), May 1990, pp. 584-586
- [leh94] LEHMAN, M.M.: 'Feedback in the Software Process', Keynote Address, CSR Eleventh Annual Wrksh. on Softw. Ev. - Models and Metrics. Dublin, 7-9th Sep. 1994. Also in Info. and Softw. Tech., spec. iss. on Softw. Maint., v. 38, n. 11, 1996, Elsevier, 1996, pp. 681 - 686
- [leh96] LEHMAN, M.M. and STENNING, V.: 'FEAST/1: Case for Support.' Department of Computing. Imperial College, March 1996, available at http://www.doc.ic.ac.uk/~mml/feast
- [leh97] LEHMAN, M.M.: 'Laws of Software Evolution Revisited'. Proc. EWSPT'96, Nancy, 9-11 October 1996, LNCS 1149, Springer Verlag, 1997, pp. 108-124
- [leh98a] LEHMAN, M.M.: 'FEAST/2: Case for Support.' Department of Computing, Imperial College, July 1998, available at http://www.doc.ic.ac.uk/~mml/feast
- [leh98b] LEHMAN, M.M., PERRY, D.E. and RAMIL, J.F.: 'On Evidence Supporting the FEAST Hypothesis and the Laws of Software Evolution'. Proc. Metrics'98, Bethesda, Maryland, 20-21 November 1998, pp. 84 - 88
- [leh00a] LEHMAN, M.M.: 'Rules and Tools for Software Evolution Planning and Management', in [fesbp]; rev. version with J.F. Ramil in Annals of Softw. Eng., spec. iss. on Softw. Managmt., v. 11, 2001
- [leh00b] LEHMAN, M.M. and RAMIL, J.F.: 'Towards a Theory of Software Evolution and its Practical Impact'. Invited Talk, Proceedings Intl. Symposium on Principles of Softw. Evolution, ISPSE 2000, 1-2 Nov, Kanazawa, Japan, pp. 2 - 11
- [leh00c] LEHMAN, M.M.: 'TheSE- An Approach to a Theory of Software Evolution', project proposal, Dept. of Computing, Imperial College, Dec. 2000, pp. 9
- [leh00d] LEHMAN, M.M. and RAMIL, J.F.: 'Software Evolution in an Era of Component Based Software Engineering', IEE Proceedings - Software, v. 147, n. 6, Dec. (2000), pp. 249 - 255, earlier version as Technical Report 98/8, Imperial College, London, Jun. 1998

References (cont.)

- [leh01a] LEHMAN, M.M. and RAMIL, J.F.: 'Software Evolution', inv. keynote lect., IWPSE 2001, Vienna, Sept. 10-11, a revised and extended version of an article to appear in Marciniak J. (ed.), Encyclopedia of Software Engineering, 2nd. Ed., Wiley, 2002
- [leh01b] LEHMAN, M.M. and RAMIL, J.F.: 'EPiCS: Evolution Phenomenology in Component-intensive Software', proposal draft, Dept. of Comp., Imp. Col., Aug. 2001
- [mck99] McKINNEY, D.: 'Impact of Commercial Off-The-Shelf (COTS) Software on the Interface Between Systems and Software Engineering.' Invited Talk, Panel on COTS Integration, Proc. ICSE'99, 16-22 May 1999, Los Angeles, CA, pp. 627-8
- [mor00] MORISIO, M., et al: 'Investigating and Improving a COTS-Based Software Development Process', Proc. ICSE 22, 4-11 June 2000, Limerick, Ireland, pp. 32 - 41
- [myr99] MYRTVEIT, I. and STENSRUD, C.: 'Benchmarking COTS Projects Using Data Envelope Analysis', Metrics 99, Boca Raton, FL., pp. 269 - 278
- [nau69] NAUR, P. and RANDELL, B. (eds.): 'Software Engineering'. report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 October 1968, January 1969, 231 pps.
- [nus97] NUSEIBEH, B.: 'ARIANNE 5 Who Dunnit?', IEEE Software, May/June 1997, pp. 15-16
- [prosim] PROSIM 2000: 'Int. Workshop on Software Process Simulation Modelling', 12 - 14 July 2000, Imp. Col., London
- [ram01] RAMIL, J.F., LEHMAN, M.M., and SANDLER, U.: 'An Approach to Modelling Long-Term Growth Trends in Large Software Systems', submitted to ICSM 2001, 6-10 November, Florence, Italy
- [raj00] RAJLICH, V.T. and BENNETT, K.H.: 'A Staged Model for the Software Life Cycle', Computer, July 2000, pp. 66 - 71
- [sea99] SEAMAN, C: 'Qualitative Methods in Empirical Studies of Software Engineering', IEEE Trans. on Softw. Eng, vol. 25, n. 4, July/Aug. 1999, pp. 557 - 572
- [sen90] SENGE, P: 'The Fifth Discipline - The Art & Practice of The Learning Organisation', Currency/Doubleday, NY, 1990, pp. 423
- [soce] SOCE 2000:'Workshop on Software and Organisation Co-evolution'. 12-13 July 2000. Imp. Col., London
- [tur87] TURSKI, W.M. and MAIBAUM, T.: 'The Specification of Computer Programs', Addison Wesley, London, 1987, p. 278)
- [tur96] TURSKI, W.M.: 'Reference Model for Smooth Growth of Software Systems,' IEEE Trans. on Soft. Eng. 22, (8), August 1996, pp. 599-600
- [voa99] VOAS, J.M.: 'Disposable Information Systems: The Future of Software Maintenance?,' Journal of Softw. Maint: Res. Pract., 1999, 11, pp. 143-150
- [wol85] WOLSTENHOLME, E.F.: 'A Methodology for Qualitative System Dynamics', in The 1985 Int. Conf. of the System Dynamics Society, 1985
- [wu00] WU, Y., PAN, D. and CHEN, M.H.: 'Techniques of Maintaining Evolving Component-based Software', ICSM 2000, 11-14 Oct., San Jose, CA, pp. 236 - 246
- [zur68] ZURCHER, F.W. and RANDELL, B: 'Iterative Multi-Level Modelling - A Methodology for Computer System Design', Info. Proc. 67, Proc. IFIP Congr. 1968, Edinburgh, Aug. 1968, pp. D138 - 142